

The Complexity of Manipulative Attacks in Nearly Single-Peaked Electorates*

Piotr Faliszewski

Department of Computer Science
AGH Univ. of Science and Technology
Kraków, Poland

Edith Hemaspaandra

Department of Computer Science
Rochester Inst. of Technology
Rochester, NY, USA

Lane A. Hemaspaandra

Department of Computer Science
Univ. of Rochester
Rochester, NY, USA

May 25, 2011, revised July 6, 2012

Abstract

Many electoral bribery, control, and manipulation problems (which we will refer to in general as “manipulative actions” problems) are NP-hard in the general case. It has recently been noted that many of these problems fall into polynomial time if the electorate is single-peaked (i.e., is polarized along some axis/issue). However, real-world electorates are not truly single-peaked. There are usually some mavericks, and so real-world electorates tend to merely be nearly single-peaked. This paper studies the complexity of manipulative-action algorithms for elections over nearly single-peaked electorates, for various notions of nearness and various election systems. We provide instances where even one maverick jumps the manipulative-action complexity up to NP-hardness, but we also provide many instances where a reasonable number of mavericks can be tolerated without increasing the manipulative-action complexity.

1 Introduction

Elections are a model of collective decision-making so central in human and multiagent-systems contexts—ranging from planning to collaborative filtering to reducing web spam—that it is natural to want to get a handle on the computational difficulty of finding whether manipulative actions can obtain a given outcome (see the survey [22]). A recent line of work started by Walsh [37, 20, 4] has looked at the extent to which NP-hardness results for the complexity of manipulative actions (bribery, control, and manipulation) may evaporate when one focuses on electorates that are (unidimensional) single-peaked, a central social-science model of electoral behavior. That model basically views society as polarized along some (perhaps hidden) issue or axis. However, real-world elections are unlikely to be perfectly single-peaked. Rather, they are merely very close to being single-peaked, a notion that was recently raised in a computational context by Conitzer [7] and Escoffier et al. [16]. There will almost always be a few mavericks, who vote based on

*Also appears as URCS-TR-2011-968.

some reason having nothing to do with the societal axis. For example, in recent US presidential primary and final elections, there was much discussion of whether some voters would vote not based on the political positioning of the candidates but rather based on the candidates’ religion, race, or gender. In this paper, we most centrally study whether the evaporation of complexity results that often holds for single-peaked electorates will also occur in nearly single-peaked electorates. We prove that often the answer is yes, and sometimes the answer is no. We defer to Section 6 our discussion of previous and related work.

Among the contributions of our paper are the following.

- Most centrally, we show that in many control and bribery settings, a reasonable number of mavericks (voters whose votes are not consistent with the societal axis) can be handled. In such cases, the “complexity-shield evaporation” results of the earlier work can now be declared free from the worry that the results might hold only for perfect single-peakedness.
- We give settings, for example 3-candidate Borda and 3-candidate veto, in which even one maverick raises the (constructive coalition weighted) manipulation complexity from P to NP-hardness.
- For all scoring systems of the form $(\alpha_1, \alpha_2, \alpha_3)$, $\alpha_2 \neq \alpha_3$, we provide a dichotomy theorem determining when the (constructive coalition weighted) manipulation problem is in P and when it is NP-complete, for so-called single-caved societies.
- We show cases where the price of mavericity is paid in nondeterminism—cases where, for each k , we prove the control problem for societies with $\mathcal{O}(\log^k n)$ mavericks to be in complexity class β_k , the k th level of the limited nondeterminism hierarchy of Kintala and Fisher [31].

This paper touches on bribery, control, and manipulation, discusses various election systems and notions of nearness to single-peaked, and gives both polynomial-time attack results and NP-hardness results. It thus is not surprising that the proofs vary broadly in their techniques and approaches; we have no single approach that covers this entire range of cases. Almost all of our proofs are relegated to the appendix.

2 Preliminaries

In this section we give intuitive descriptions of the problems that we study. More detailed coverage, and discussion of the motivations and limitations of the models, can be found in the various bibliography entries, including, for example, Faliszewski et al. [19, 22]. We have also included formal definitions in the appendix. As is standard, throughout this paper the terms “NP-hard”/“NP-hardness” will refer to polynomial-time many-one “NP-hard”/“NP-hardness.”

Elections An election $E = (C, V)$ consists of a finite candidate set C and a finite collection V of votes over the candidates. V is a list of entries, one per voter, with each entry containing a linear (i.e., tie-free total) ordering of the candidates (except for approval elections where each vote is a $\|C\|$ -long 0-1 vector denoting disapproval/approval of each candidate).¹ In plurality elections, whichever candidate gets the most top-of-the-preference-order votes wins. Each vector $(\alpha_1, \dots, \alpha_k)$, $\alpha_i \in \mathbb{N}$, $\alpha_1 \geq \dots \geq \alpha_k \geq 0$, defines a k -candidate scoring protocol election, in which each voter’s i th favorite candidate gets α_i points, and whichever candidate

¹Throughout this paper V , though input as a list (one ballot per voter), typically functions as a multiset. When dealing with voter sets, we use terms such as set/subset to mean multiset/submultiset (although we will typically just say collection), and we use set-bracket notation to, for that case, mean multisets, e.g., if v is a preference order, w is a preference order, $V = \{v, v\}$, and $W = \{v, w\}$, then $V \cup W = \{v, v, v, w\}$.

gets the most points wins. k -candidate veto is defined by the vector $(\overbrace{1, \dots, 1}^{k-1}, 0)$, and k -candidate Borda is defined by the vector $(k-1, k-2, \dots, 0)$. In approval elections, whichever candidate is approved of by the most voters wins. In all the systems just mentioned, if candidates tie for the highest number of points, those tying for highest are all considered winners. In Condorcet elections, a candidate wins if he or she strictly beats every other candidate in pairwise head-on-head votes.

Attacks Each of the election problems is defined based on an election and some additional parameters. In constructive coalition weighted manipulation (CCWM), the input is a set of nonmanipulative voters (having weights and preferences over the candidates), a list of the weights of the manipulative voters, and which candidate p the manipulators wish to be a winner. (Input) instances are in the set exactly if there is a set of votes the manipulators can cast to make p a winner (under the given election system). In “bribery,” all voters have preferences, and our input is an election, a candidate p , and a bound K on how many voters can be bribed. Instances are in the set if there is a way of changing the preferences of at most K voters that makes p a winner. (We will briefly mention a number of variations of bribery. “Weighted” means the voters have weights, “\$” means voters have individual prices (and K becomes a bound on the amount that can be spent seeking to make p win). For approval voting, “negative” bribery means a bribe cannot change someone from disapproving of p to approving of p , and “strongnegative” bribery means every bribed person must end up disapproving of p . In negative bribery, one can only help p in subtle, indirect ways. For plurality, the negative notion is similar, see Faliszewski et al. [18] or our appendix.) In control (of the four types we will discuss), the input is an election, the candidate p one wants to be a winner, and a parameter K limiting how many actors one can influence in the designated way. Our four types of control will be adding voters (CCAV), deleting voters (CCDV), adding candidates (CCAC), and deleting candidates (CCDC). Each of those four problems is defined as the collection of inputs on which using at most K actions of the designated type (e.g., adding at most K voters) suffices to make p a winner. For CCAV an additional part of the input is a pool of potential additional voters (and their preferences over the candidates). For CCAC an additional part of the input is a pool of potential additional candidates, and all voters have preferences over the set of all initial and potential-additional candidates. (Some early papers on control focused on making p be the one and only winner, but we follow the more recent approach of focusing on making p become a winner. The older results for the former case that we cite here are known in the literature, or were verified for this paper by us, to also hold for the latter case.) Control loosely models such real-world activities as get-out-the-vote drives, targeted advertising, and voter suppression.

Each of our algorithms for manipulation, bribery, and control not only gives a *yes/no* answer for the decision variant of the problem, but also can be made to produce a successful manipulative action if the answer is *yes*.

Having algorithms for such tasks as bribery and control isn’t inherently a “bad” or unethical thing. For example, bribery and control algorithms are valuable tools for actors (party chairs, campaign managers, etc.) who are trying to most effectively use their resources.

(Nearly) Single-peakedness A collection V of votes (cast as linear orders) is said to be single-peaked exactly if there is a linear order L over the candidate set such that for each triple of candidates, c_1, c_2, c_3 , it holds that if $c_1 L c_2 L c_3 \vee c_3 L c_2 L c_1$, then $(\forall v \in V)[c_1 P_v c_2 \implies c_2 P_v c_3]$, where $a P_v b$ means that voter v prefers a to b . This notion was first created by Black more than half a century ago, and is one of the most important concepts in political science. Loosely put, the notion is motivated as follows. Imagine that on some issue, for example what the tax rate should be for the richest Americans, each person has a utility curve that on a (perhaps empty) initial part is nondecreasing and then on the (perhaps empty) rest is nonincreasing. Suppose the candidates are spread along the tax-rate axis as to their positions, with no two on

top of each other. The set of preferences that can be supported among them by curves of the mentioned sort on which there are no ties among candidates in utility are precisely the single-peaked vote ensembles. Note that different voters can have different peaks/plateaus and different curves, e.g., if both Alice and Bob think 40 percent is the ideal top tax rate, it is completely legal for Alice to prefer 30 percent to 50 percent and Bob to prefer 50 percent to 30 percent. There is extensive political science literature on single-peaked voting’s naturalness, ranging from conceptual discussions to empirical studies of actual US political elections (with few candidates) showing that most voters are single-peaked with respect to left-right political spectrum, and has been described as “*the canonical setting for models of political institutions*” [25]. If in the definition of single-peaked one replaces the final “forall” with this one, $(\forall v \in V)[c_2 P_v c_1 \implies c_3 P_v c_2]$, one defines the closely related notion of single-caved preferences, which we will also study. For approval ballots, a vote set V is said to be single-peaked if there is a linear order L such that for each voter v , all candidates that v approves of (if any) form an adjacent block in L .

In all our manipulative action problems about single-peaked and nearly single-peaked societies, we will follow Walsh’s model, which is that the societal order, L , is part of the input. (See the earlier papers for extensive discussion of why this is a reasonable model.)

In this paper, we will primarily focus on elections whose voters are “nearly” single-peaked, under the following notions of nearness. Our “maverick” notions apply to both voting by approval ballots and voting by linear orders; our other notions are specific to voting by linear orders. We will say an election is over a k -maverick-SP society (equivalently, a k -maverick-SP electorate) if all but k of the voters are consistent with (in the sense of single-peakedness; this does not mean identical to) the societal order L . That is, we allow up to k mavericks. We will speak of $f(\cdot)$ -maverick-SP societies when this usage and the type of f ’s argument(s) is clear from context (f ’s argument(s) will typically be the size of the election instance or some parameters of the election, e.g., the number of candidates or the number of voters).

Also, we will prove a number of results that state that “PROBLEM for ELECTION-SYSTEM over log-maverick-SP societies is in P”; this is a shorthand for the claim that for each function f (that is computable in time polynomial in the size of the input—which is roughly $\|V\| \|C\| \log \|C\|$ for the election (C, V) itself plus whatever space is taken by other parameters—and to avoid possible technical problems, we should assume f is nondecreasing) whose *value* is $\mathcal{O}(\log(\text{ProblemInputSize}))$, it holds that “PROBLEM for ELECTION-SYSTEM over f -maverick-SP societies is in P,” where the argument to f is the input size of the problem. An election is over a (k, k') -swoon-SP society if each voter has the property that if one removes the voter’s k favorite and k' least favorite candidates from the voter’s preference order, the resulting order is consistent with societal order L after removing those same candidates from L . We will use swoon-SP as a shorthand for $(1, 0)$ -swoon-SP, as we will not study other swoon values in this paper. In swoon-SP, each person may have as her or his favorite some candidate chosen due to some personal passion (such as hairstyle or religion), but all the rest of that person’s vote must be consistent with the societal polarization. An election is over a Dodgson $_k$ -SP society if for each voter some at-most- k sequential exchanges of adjacent candidates in his or her order make the vote consistent with the societal order L . An election is over a PerceptionFlip $_k$ -SP society if, for each voter, there is some series of at most k sequential exchanges of adjacent candidates *in the societal order* L after which the voter’s vote is consistent with L . This models each voter being consistent with that voter’s humanly blurred view of the societal order.

Some model details follow. For control by adding voters for maverick-SP societies, the total number of mavericks in the initial voter set and the pool of potential additional voters is what the maverick bound limits. For manipulation, nonmanipulators as well as manipulators can be mavericks, and we bound the total number of mavericks. For bribery involving $f(\cdot)$ -maverick-SP societies, we will consider both the “standard” model and the “marked” model. In the standard model, the $f(\text{ProblemInputSize})$ limit on the

number of mavericks must hold both for the input and for the voter set after the bribing is done; anyone may be bribed and bribes can create mavericks and can make mavericks become nonmavericks. In the marked model, each voter has a flag saying whether or not he or she can cast a maverick vote (we will call that being “maverick-enabled”). The at most $f(\text{ProblemInputSize})$ voters with the maverick-enabled flag may (subject to the other constraints of the bribery problem such as total number of bribes) be bribed in any way, and so may legally cross in either direction between consistency and inconsistency with the societal ordering. All non-maverick-enabled voters must be consistent with societal order L both before and after the bribing, although they too can be bribed (again, subject to the problem’s other constraints such as total number of bribes).

For single-peaked electorates, “median voting” (in which the candidate wins who on the societal axis is preferred by the “median voter”) is known to be strategy-proof, i.e., a voter never benefits from misrepresenting his or her preferences. It might seem tempting to conclude from that that all elections on single-peaked societies “should” use median voting, and that we thus need not discuss single-peaked (or perhaps even nearly single-peaked) elections with respect to other voting systems, such as plurality, veto, etc. But that temptation should be resisted. First, median voting’s strategy-proofness regards manipulation, not control or bribery. Second, even in real-world political elections broadly viewed as being (nearly) single-peaked, it simply is not the case that median voting is used. People, for whatever reasons of history and comfort, use such systems as plurality, approval, and so on for such elections. And so algorithms for those systems are worth studying. Third, for manipulation of nearly single-peaked electorates, strategy-proofness does not even hold. And although for them indeed only the mavericks can have an incentive to lie, that doesn’t mean that the outcome won’t be utterly distorted even by a single maverick. There are arbitrarily large electorates, having just one maverick, where that maverick can change the winner from being the median one to instead being a candidate on the outer extreme of the societal order.

3 Manipulation

This paper’s sections on control and bribery focus on, and provide many examples of, settings where not just the single-peaked case but even the nearly single-peaked cases have polynomial-time algorithms. Regarding manipulation, the results are more sharply varied.

We show that NP-hardness holds for a rich class of scoring protocols, in the presence of even one maverick. (When $\alpha_2 = \alpha_3$ the system is either equivalent to plurality or is a trivial system where everyone always is a winner. These cases are easily seen to be in P.) Recall from Section 2 the meaning of “ $(\alpha_1, \alpha_2, \alpha_3)$ elections,” namely, scoring protocol elections using the vector $(\alpha_1, \alpha_2, \alpha_3)$.

Theorem 3.1. *For each $\alpha_1 \geq \alpha_2 > \alpha_3$, CCWM for $(\alpha_1, \alpha_2, \alpha_3)$ elections over 1-maverick-SP societies is NP-complete.*

We point out that this theorem is of the same form as that for the general case (see [8, 28, 35]). However, the proofs for the general case do not work in our case, since those proofs construct elections with at least two mavericks.

In the general case (i.e., no single-peakedness is required), the above cases also are NP-complete [8, 28, 35], so allowing a one-maverick single-peaked society is jumping us up to the same level of complexity that holds in the general case here. In contrast, for SP societies (without mavericks), 3-candidate CCWM is NP-complete when $(\alpha_1 - \alpha_3) > 2(\alpha_2 - \alpha_3) > 0$ and is in P otherwise [23]. So, in particular, 3-candidate veto and 3-candidate Borda elections are in P for the SP (single-peaked) case, but are already NP-complete for SP with one maverick allowed.

Does allowing one maverick always raise the CCWM complexity? No, as the following theorem shows. (The $k = 0$ case follows from Faliszewski et al. [23].)

Theorem 3.2. *For each $k \geq 0$ and $m \geq k + 3$, CCWM for m -candidate veto elections over k -maverick-SP societies is in P.*

In contrast, all of Theorem 3.2's cases are well-known to be NP-complete in the general case [8]. Still, the contrast is a bit fragile. For example, although the above theorem shows that CCWM for $(1, 1, 1, 1, 0)$ elections over 2-maverick-SP societies is in P, we prove below that CCWM for $(1, 1, 1, 0)$ elections over 2-maverick-SP societies is NP-complete. Note also that this theorem gives an example where 4-candidate veto elections are NP-complete but 5-candidate veto elections are in P, in contrast with the behavior that one often expects regarding NP-completeness and parameters, namely, one might expect that increasing the number of candidates wouldn't lower the complexity. (However, see Faliszewski et al. [23] for another example of this unusual behavior.)

Theorem 3.3. *For each $k \geq 0$ and $m \geq 3$ such that $m \leq k + 2$, CCWM for m -candidate veto elections over k -maverick-SP societies is NP-complete.*

Theorems 3.2 and 3.3 also show that for any number of mavericks, there exists a voting system such that CCWM is easy for up to that number of mavericks, and hard for more mavericks.

Corollary 3.4. *Let $k \geq 0$. For all $\ell \geq 0$, CCWM for $k + 3$ -candidate veto elections over ℓ -maverick-SP societies is in P if $\ell \leq k$ and NP-complete otherwise.*

Let us now turn from the maverick notion of nearness to single-peakedness, and look at the “swoon” notion, in which, recall, each voter must be consistent with the societal ordering (minus the voter's first-choice candidate) when one removes from the voter's preference list the voter's first-choice candidate. We will still mostly focus on the case of veto elections. For three candidates (see Observation 3.6) and four candidates we have NP-completeness, and for five or more candidates we have membership in P.

Theorem 3.5. *For each $m \geq 5$, CCWM for m -candidate veto elections in swoon-SP societies is in P. For $m \in \{3, 4\}$, this problem is NP-complete.*

Observation 3.6. *Every 3-candidate election is a swoon-SP election and a Dodgson₁-SP election and so all complexity results for 3-candidate elections in the general case also hold for swoon-SP elections and Dodgson₁-SP elections.*

Complexity results for general elections do not always hold for swoon-SP elections or for Dodgson₁-SP elections. For example, for $m \geq 5$, CCWM for m -veto elections is NP-complete in the general case, but in P for swoon-SP societies (Theorem 3.5) and Dodgson₁-SP societies (Theorem 3.7).

Theorem 3.7. *For each $m \geq 5$, CCWM for m -candidate veto elections in Dodgson₁-SP societies is in P. For $m \in \{3, 4\}$, this problem is NP-complete.*

We conclude this section with a brief comment about single-caved electorates. (We remind the reader that single-caved is not a “nearness to SP” notion, but rather is in some sense a mirror-sibling.) For scoring vectors $(\alpha_1, \alpha_2, \alpha_3)$, the known CCWM dichotomy result for single-peaked electorates is that if $\alpha_1 - \alpha_3 > 2(\alpha_2 - \alpha_3)$ then the problem is NP-complete and otherwise the problem is in P. For single-caved, the opposite holds for each case that is not in P in the general case.

Theorem 3.8. *For each $\alpha_1 \geq \alpha_2 > \alpha_3$, CCWM for $(\alpha_1, \alpha_2, \alpha_3)$ elections over single-caved societies is NP-complete if $(\alpha_1 - \alpha_3) \leq 2(\alpha_2 - \alpha_3)$ and otherwise is in P.*

4 Control

The very first results of Faliszewski et al. [23] showing that NP-complete general-case control results can simplify to P results for single-peaked electorates were for constructive control by adding voters and for constructive control by deleting voters, for approval elections. We show that each of those results can be reestablished even in the presence of logarithmically many mavericks. (Indeed, we mention in passing that even if the attacker is allowed to simultaneously both add and delete voters—so-called “AV+DV” multimode control in the recent model that allows simultaneous attacks [17]—the complexity of planning an optimal attack still remains polynomial-time even with logarithmically many mavericks.)

Theorem 4.1. *CCAV and CCDV for approval elections over log-maverick-SP societies are each in P. For CCAV, the complexity remains in P even for the case where no limit is imposed on the number of mavericks in the initial voter set, and the number of mavericks in the set of potential additional voters is logarithmically bounded (in the overall problem input size).²*

Although we will soon prove a more general result, we start by proving this result directly. We do so both as that will make the more general result clearer, and as P-time results are the core focus of this paper. Our proof involves the “demaverickification” of the society, in order to allow us to exploit the power of single-peakedness. By doing so, our proof establishes that there is a polynomial-time disjunctive truth-table reduction (see Ladner et al. [32] for the formal definition of \leq_{dt}^P , but one does not need to know that to follow our proof) to the single-peaked case. In this version of this paper, the proof of Theorem 4.1 is located in Appendix C, and we assume that the interested reader will now read it (doing so is not required and probably not even a good idea on a first reading, but such reading will make clearer the next paragraph, which briefly refers to the algorithm within that proof). Now, before we move on to other election systems, let us pause to wonder whether Theorem 4.1 is just the tip of an iceberg, and is in fact hiding some broader connection between number of mavericks and computational complexity theory. We won’t give this type of discussion for all, or even most, of our theorems. But it is worthwhile to, since this is our first control result, look here at what holds. And what holds is that Theorem 4.1 is indeed in some sense the tip of an iceberg. However, it is an iceberg whose tip is its most interesting part, since it gives the part that admits polynomial-time attacks.

Still, the rest of the iceberg brings out an interesting connection between maverick frequency and non-determinism. Let us think again of the proof we just saw. It worked by sequentially generating each member of the powerset of a logarithmic-sized set (call it Q). And we did that, naturally enough, in polynomial time. However, note that we could also have done it with nondeterminism. We can nondeterministically guess for each member of Q whether or not it will be added (for CCAV) or deleted (for CCDV). And then after that nondeterministic guess, we for the CCAV case do the demaverickification presented in the above theorem’s proof and for the CCDV case do the deletable/nondeletable marking, and then we run the polynomial-time algorithms for the single-peaked approval-voting CCAV and the approval-voting CCDV (with deletable/nondeletable flag, and all mavericks—those not consistent with the societal ordering—being nondeletable) cases. It is easy to see that above proof argument works fine with the

²By that last part, we mean precisely the definition—including its various restrictions on the complexity of the function—of the notion of log-maverick-SP, except with the limit being placed just on the number of mavericks in the additional voter set. Formally put, to avoid any possibility of ambiguity or confusion, we mean that for each function f (that is computable in time polynomial in the size of the input; and to avoid possible technical problems, we require that f be nondecreasing) whose *value* is $\mathcal{O}(\log(\text{ProblemInputSize}))$, it holds that “CCAV for approval elections over inputs for which (as is standard in our model regarding SP and nearly-SP cases, a societal ordering of single-peakedness is given a part of the input, and) at most $f(\text{ProblemInputSize})$ of the additional voters are inconsistent with the societal ordering is in P.”

change to nondeterminism. Indeed, the reason Theorem 4.1 is about “P” is because sequentially handling $\mathcal{O}(\log(\text{ProblemInputSize}))$ nondeterministic bits can be done in polynomial time.

So, what underlies the above theorem are the following results that say that frequency of mavericks in one’s society exacts a price, in nondeterminism. (We here are proving just an upper bound, but we conjecture that the connection is quite tight—that commonality of wild voter behavior is very closely connected with nondeterminism.) To state the results, we need to briefly introduce some notions from complexity theory. Complexity theorists often separate out the weighing of differing resources, putting bounds on each. The only such class we need here is the class of languages that can be accepted in time $t(n)$ on machines using $g(n)$ bits of nondeterminism, which is typically denoted $\text{NONDET-TIME}[g(n), t(n)]$. The most widely known such classes are those of the limited nondeterminism hierarchy, known as the beta hierarchy, of Kintala and Fisher [31] (see also [9] and the survey [27]). β_k is the class of sets that can be accepted in polynomial time on machines that use $\mathcal{O}(\log^k n)$ bits of nondeterminism: $\beta_k = \{L \mid (\exists \text{ polynomial } t(n))(\exists g(n) \in \mathcal{O}(\log^k n))[L \in \text{NONDET-TIME}[g(n), t(n)]]\}$, or for short, $\beta_k = \text{NONDET-TIME}[\mathcal{O}(\log^k n), \text{poly}]$. Of course, $\beta_0 = \beta_1 = \text{P}$. We can now state our result, which says that frequency of mavericity is paid for in nondeterminism.

Theorem 4.2. *CCAV and CCDV for approval elections over $f(\cdot)$ -maverick-SP societies are each in $\text{NONDET-TIME}[f(\text{ProblemInputSize}), \text{poly}]$. For CCAV, the complexity remains in $\text{NONDET-TIME}[f(\text{ProblemInputSize}), \text{poly}]$ even for the case where no limit is imposed on the number of mavericks in the initial voter set, and the number of mavericks in the set of potential additional voters is $f(\cdot)$ -bounded (in the overall problem input size).*

Corollary 4.3. *For each natural number k , CCAV and CCDV for approval elections over $\mathcal{O}(\log^k n)$ -maverick-SP societies are each in β_k . For CCAV, the complexity remains in β_k even for the case where no limit is imposed on the number of mavericks in the initial voter set, and the number of mavericks in the set of potential additional voters is $\mathcal{O}(\log^k n)$ -bounded (in the overall problem input size).*

Theorem 4.1 follows from the $k = 1$ case of this more general corollary. Now let us turn from our particularly detailed discussion of CCAV and CCDV for approval voting, and let us look at other election systems.

For Condorcet elections, both CCAV and CCDV are known to be NP-complete in the general case [3] but to be in P for single-peaked electorates [4]. For Condorcet, results analogous to those for approval under $f(\cdot)$ -maverick-SP societies hold.

Theorem 4.4. *CCAV and CCDV for Condorcet elections over $f(\cdot)$ -maverick-SP societies are each in $\text{NONDET-TIME}[f(\text{ProblemInputSize}), \text{poly}]$. For CCAV, the complexity remains in $\text{NONDET-TIME}[f(\text{ProblemInputSize}), \text{poly}]$ even for the case where no limit is imposed on the number of mavericks in the initial voter set, and the number of mavericks in the set of potential additional voters is $f(\cdot)$ -bounded (in the overall problem input size).*

Corollary 4.5. *For each natural number k , CCAV and CCDV for Condorcet elections over $\mathcal{O}(\log^k n)$ -maverick-SP societies are each in β_k . For CCAV, the complexity remains in β_k even for the case where no limit is imposed on the number of mavericks in the initial voter set, and the number of mavericks in the set of potential additional voters is $\mathcal{O}(\log^k n)$ -bounded (in the overall problem input size).*

For plurality, the most important of systems, CCAV and CCDV are known to be in P for the general case [3], so there is no need to seek a maverick result there. However, CCAC and CCDC are both known to

be NP-complete in the general case and in P in the single-peaked case. For both of those, a constant number of mavericks can be handled.

Theorem 4.6. *For each k , CCAC and CCDC for plurality over k -maverick-SP societies are in P.*

As in the case of Theorem 4.1, the idea of the algorithm is to (polynomial-time disjunctively truth-table) reduce to the single-peaked case. However, here the mavericks require a more involved brute-force search and thus we can only handle a constant number of them.

Unfortunately, swooning cannot be handled at all (unless $P = NP$, of course). Also, allowing the number of mavericks to be some root of the input size cannot be handled.

Theorem 4.7. *CCAC and CCDC for plurality elections over swoon-SP societies are NP-complete.*

Theorem 4.8. *For each $\varepsilon > 0$, CCAC and CCDC for plurality elections over I^ε -maverick-SP societies are NP-complete, where I denotes the input size.*

For the Dodgson and PerceptionFlip notions of nearness to single-peakedness, we can prove that allowing a constant number of adjacent-swaps (for each voter, separately, in the appropriate structure) still leaves the CCAC and CCDC problems in P. (We mention in passing that the following result holds not just for constructive control, but even for the concept, which we are not focusing on in this paper, of destructive control, in which one's goal is just to preclude a certain candidate from winning.)

Theorem 4.9. *For each k , CCAC and CCDC for plurality elections are in P for Dodgson_k -SP societies and for PerceptionFlip_k -SP societies.*

Our algorithm exploits the local nature of adding/deleting candidates in single-peaked plurality elections. Formally, Faliszewski et al. [23] made the following observation.

Lemma 4.10 (Lemma 3.4 of [23]). *Let (C, V) be an election where $C = \{c_1, \dots, c_m\}$ is a set of candidates, V is a collection of voters whose preferences are single-peaked with respect to societal axis L , and where $c_1 L c_2 L \dots L c_m$. Within plurality, if $m \geq 2$ then*

1. $\text{score}_{(C, V)}(c_1) = \text{score}_{(\{c_1, c_2\}, V)}(c_1)$,
2. for each i , $2 \leq i \leq m - 1$, $\text{score}_{(C, V)}(c_i) = \text{score}_{(\{c_{i-1}, c_i, c_{i+1}\}, V)}(c_i)$, and
3. $\text{score}_{(C, V)}(c_m) = \text{score}_{(\{c_{m-1}, c_m\}, V)}(c_m)$.

It turns out that this local structure, slightly distorted, still occurs in Dodgson_k -SP societies and in PerceptionFlip_k -SP societies. Thus, our strategy for proving Theorem 4.9 is to first formally define what we mean by a distorted variant of the above lemma, and then to adapt the CCAC and CCDC algorithms of Faliszewski et al. [23] for single-peaked societies to the distorted setting.

Definition 4.11. *Let $C = \{c_1, \dots, c_m\}$ be a set of candidates and let L be a linear order over C (the societal axis) such that $c_1 L c_2 L \dots L c_m$. For each $c_i \in C$ and each nonnegative integer k , $0 \leq k \leq m - 1$, we define $N(L, C, c_i, k) = \{c_j \mid |i - j| \leq k\}$. We call $N(L, C, c_i, k)$ the k -radius neighborhood of c_i with respect to C and L .*

Definition 4.12. *Let $E = (C, V)$ be a plurality election, let L be a linear order over C (the societal axis), and let k be a positive integer. We say that E is k -local with respect to L if for each $c \in C$ and each $C' \subseteq C$ such that $c \in C'$ it holds that $\text{score}_{(N(L, C', c, k), V)}(c) = \text{score}_{(C', V)}(c)$.*

In particular, the proof of Lemma 4.10 (given in [23]) shows that single-peaked plurality elections are 1-local with respect to the societal axis. We extend this result to Dodgson_k-SP societies and to PerceptionFlip_k-SP societies.

Lemma 4.13. *Let k be a positive integer, let $E = (C, V)$ be a plurality election, and let L be some linear order over C (the societal axis).*

- (a) *If E is Dodgson_k-SP with respect to L then E is $(k+1)$ -local with respect to L .*
- (b) *If E is PerceptionFlip_k-SP with respect to L then E is $(k+1)$ -local with respect to L .*

Proof. Cases (a) and (b) are similar but not identical and thus we will handle each of them separately.

Case (a) Let $E = (C, V)$ be a Dodgson_k-SP election, where $C = \{c_1, \dots, c_m\}$ and where the societal axis L is such that $c_1 L c_2 L \dots L c_m$. Since for every $C' \subseteq C$, (C', V) is Dodgson_k-SP with respect to L and since E was chosen arbitrarily, it suffices to show that for each candidate $c_i \in C$ it holds that $\text{score}_{(N(L, C, c_i, k+1), V)}(c_i) = \text{score}_{(C, V)}(c_i)$.

Fix a candidate c_i in C and a voter v in V . By definition of Dodgson_k-SP elections, there exists a vote v' that is single-peaked with respect to L , such that v can be obtained from v' by at most k swaps. Let $C' = N(L, C, c_i, k+1)$. We will show that $\text{score}_{(C', \{v\})}(c_i) = \text{score}_{(C, \{v\})}(c_i)$.

First, if v ranks c_i on top among all candidates in C , then certainly v ranks c_i on top among candidates in C' . Thus, $\text{score}_{(C', \{v\})}(c_i) \geq \text{score}_{(C, \{v\})}(c_i)$. It remains to show that if v does not rank c_i on top among candidates in C , then v also does not rank c_i on top among candidates in C' . We consider two cases:

1. The peak of v' is not in C' . Then it is easy to verify that some candidate from C' precedes c_i in v . Otherwise, to convert v' to v one would need enough swaps for c_i to pass $k+1$ candidates from C' (either those “to the left” of c_i in L if the peak of v' was “to the left” of c_i , or those “to the right” of c_i if the peak of v' was “to the right” of c_i).
2. The peak of v' is in C' . If v 's top-ranked candidate is in C' then clearly c_i is not ranked first among C' in v . Thus, let us assume that the top ranked candidate in v is not in C' and that it is some candidate c_j . Without loss of generality, let us assume that $j > i + k + 1$ (the case where $j < i - k - 1$ is analogous). Let us also assume that the peak of v' is some candidate c_{i+s} , such that $1 \leq s \leq k+1$ (the case when $-k-1 \leq s \leq 0$ is impossible because converting v' to v requires at most k swaps). The minimal number of swaps that convert v' to a vote where c_j is ranked first is at least $(k+1) - s + 1 = k - s + 2$ (these swaps involve c_j and candidates $c_{i+s}, c_{i+s+1}, \dots, c_{i+k+1}$). The minimum number of swaps in v' that ensure that c_i is ranked ahead of c_{i+s} is at least s (these swaps involve c_i and candidates $c_{i+1}, c_{i+2}, \dots, c_{i+s}$). Thus, the minimum number of swaps of candidates in v' that ensure that c_j is ranked first and that c_i is ranked ahead of c_{i+s} is $k+2$, which is more than the allowed k swaps. Thus, this situation is impossible. As a result, some candidate from C' is ranked ahead of c_i in v .

Thus, we have shown that if c_i is not ranked first in v among the candidates from C , then c_i is not ranked first in v among the candidates from C' . Thus, $\text{score}_{(C', \{v\})}(c_i) \leq \text{score}_{(C, \{v\})}(c_i)$, and with the previously shown inequality $\text{score}_{(C', \{v\})}(c_i) \geq \text{score}_{(C, \{v\})}(c_i)$, it must be the case that $\text{score}_{(C', \{v\})}(c_i) = \text{score}_{(C, \{v\})}(c_i)$. Since v was chosen arbitrarily, we have that $\text{score}_{(C', V)}(c_i) = \text{score}_{(C, V)}(c_i)$. This completes the proof of part (a) of the theorem.

Case (b) Let $E = (C, V)$ be an election, where $C = \{c_1, \dots, c_m\}$. Let us assume, without loss of generality, that E is PerceptionFlip_k-SP via societal axis L , where $c_1 L c_2 L \dots L c_m$. Since for every $C' \subseteq C$, (C', V) is

PerceptionFlip $_k$ -SP with respect to L and since E was chosen arbitrarily, it suffices to show that for each candidate $c_i \in C$ it holds that $\text{score}_{(N(L,C,c_i,k+1),V)}(c_i) = \text{score}_{(C,V)}(c_i)$.

Let us fix a candidate c_i in C and a voter v in V . This voter's preference order is single-peaked with respect to some order L' that can be obtained from L by at most k swaps of adjacent candidates. Let us assume that $c_{j'}$ is the candidate directly preceding c_i in L' and $c_{j''}$ is the candidate directly succeeding c_i in L' (in this proof we omit the easy-to-handle cases where c_i is either the maximum or the minimum element of L').

We claim that for any $C' \subseteq C$ that includes $c_{j'}$, c_i , and $c_{j''}$, it holds that $\text{score}_{(C',\{v\})}(c_i) = \text{score}_{(C,\{v\})}(c_i)$. This is so because any voter that ranks c_i on top, ranks c_i on top irrespective of which other candidates are included. So, $\text{score}_{(C',\{v\})}(c_i) \geq \text{score}_{(C,\{v\})}(c_i)$. On the other hand, by Lemma 3.4 of [23], $\text{score}_{(\{c_{j'},c_i,c_{j''}\},\{v\})}(c_i) = \text{score}_{(C,\{v\})}(c_i)$. Thus, any voter that does not rank c_i on top, given a choice between c_i , $c_{j'}$, and $c_{j''}$ ranks one of $c_{j'}$, $c_{j''}$ on top. It is easy to see that $\{c_i, c_{j'}, c_{j''}\} \subseteq N(L, C, c_i, k+1)$, and so, $\text{score}_{(C',\{v\})}(c_i) \leq \text{score}_{(C,\{v\})}(c_i)$. Thus, $\text{score}_{(C',\{v\})}(c_i) = \text{score}_{(C,\{v\})}(c_i)$ and since we picked v arbitrarily, $\text{score}_{(C',V)}(c_i) \leq \text{score}_{(C,V)}(c_i)$. The proof of case (b) is complete. \square

Now, Theorem 4.9 is a consequence of the following, more general, result.

Theorem 4.14. *For each constant k , CCAC and CCDC for plurality elections are in P for k -local elections.*

However, we do have NP-completeness if in Dodgson $_k$ -SP societies or PerceptionFlip $_k$ -SP societies we allow the parameter k to increase to $m-2$, where m is the total number of candidates involved in the election. (This many swaps allow us to, in effect, use the same technique as for swoon-SP societies.)

Theorem 4.15. *CCAC and CCDC for plurality elections are NP-complete for Dodgson $_{m-2}$ -SP societies and for PerceptionFlip $_{m-2}$ -SP societies, where m is the total number of candidates involved in the election.*

Finally, we note that for single-caved elections, CCAC and CCDC are in P for plurality.

Theorem 4.16. *CCAC and CCDC for plurality elections are in P for single-caved societies.*

5 Bribery

We now briefly look at bribery of nearly single-peaked electorates, focusing on approval elections. For all three cases—bribery, negative-bribery, and strongnegative-bribery—in which general-case NP-hardness bribery results have been shown to be in P for single-peaked societies [4], we show that the complexity remains in polynomial time even if the number of mavericks is logarithmic in the input size.

Theorem 5.1. *Bribery, negative-bribery, and strongnegative-bribery for approval elections over log-maverick-SP societies are each in P, in both the standard and the marked model.*

We mention in passing that although plurality bribery has never been discussed with respect to single-peaked (or nearly single-peaked) electorates, it is not hard to see that the two NP-complete bribery cases for plurality (plurality-weighted-bribery and plurality-weighted-negative-bribery) remain NP-complete on single-peaked electorates, in one case immediately from a theorem of, and in one case as a corollary to a proof of, Faliszewski et al. [18].

6 Related Work

Although it has roots going even further back, the study of the computational complexity of control and manipulation actions was started by a series of papers of Bartholdi, Orlin, Tovey, and Trick around 1990 [2, 1, 3] and the complexity of bribery was first studied far more recently [18]. For further references, history, context, and results regarding all of these, see the surveys [19, 22]. For example, it is known that there exist election systems that are resistant to many control attacks [14, 11, 15, 21, 29].

The four papers most related to the present one are the following. Walsh [37] insightfully raised the idea that general complexity results may change in single-peaked societies. His manipulative-action example (STV) actually provides a case where single-peakedness fails to lower manipulation complexity, but in a different context he did find a lowering of complexity for single-peakedness. The papers [20, 4] then broadly explored the effect of single-peakedness on manipulative actions. These three papers are all in the model of (perfect) single-peakedness. Conitzer [7], in the context of preference elicitation, raised and experimentally studied the issue of *nearly* single-peaked societies. Escoffier et al. [16] also discussed nearness to single-peakedness, and the papers [23, 4] both raise as open issues whether shield-evaporation (complexity) results for single-peakedness will withstand near-single-peakedness. The present paper seeks to bring the “nearly single-peaked” lens to the study of manipulative actions.

It is well-known (see Elkind et al. [10] and the references therein) that many election systems are defined, or can be equivalently defined, as selecting whichever candidate’s region of being a winner under some notion of “consensus” has a vote set that is “closest” to the input vote set, where “closest” is defined by applying some norm (e.g., sum or max) to a vector whose i th component is some notion of distance (e.g., number of adjacent-swaps to get between the votes) between the i th votes on each list. We note that, similarly, most of our notions of nearness to single-peaked can be framed as saying that the input vote set is close (in the same sense) to some vote set that is consistent with the input societal linear order. The parallel isn’t perfect, since in the former work there are multiple target regions and the minimum over them is crucially selecting the winner; and also, approaches focusing on distance typically require commutativity of the distance function, but notions of diverging from a societal order may, as a matter of human behavior, be asymmetric. (The swoon notion is asymmetric; often one can swoon from v_1 to v_2 , but not vice versa). But we mention that most of our norms/distances have already proven useful in other election contexts, and that if one finds additional norms/distances natural here, then one could study what happens under those.

This paper is focused on the line of work that looks at whether single-peakedness removes NP-hardness results about manipulative actions. *Most of our key results are cases where we show that even nearly-single-peaked elections fall to P. Our results of that sort are polynomial-time upper bounds, and so apply on all inputs.* However, a few of our results are about NP-hardness. For those, it is important to mention that NP-hardness is a worst-case theory, and so such results are just a first step on a path that one hopes may eventually reach some notion of average-case hardness. That is a long-term and difficult goal, but has not yet been proven impossible. Although many people have the impression that recent results such as Friedgut et al. [24] prove that any reasonable election system can often be manipulated, that powerful paper, for example, merely proves that the manipulation probability *cannot go very quickly to zero* asymptotically—it does not prove that the manipulation probability asymptotically does not go to zero. Other work that controls the candidates-to-voters cardinality relation has experimentally suggested stronger claims in certain settings, but is of necessity within the setting of making assumptions about the distribution of votes (see, e.g., Walsh [38]), and typically presents simulations but not theorems and proofs.

Finally, given the high hopes of many people for heuristic algorithms, it is extremely important to understand that it follows from known complexity-theoretic results that (unless shocking complexity-class

collapses occur) *no polynomial-time algorithm can come too close to accepting any NP-hard set*. In particular, if any polynomial-time heuristic correctly solves any NP-hard problem on all but a sparse (i.e., at most a polynomial number of strings of each length) set of inputs, then $P = NP$ (Schöning [36] established a precursor of this result that held for appropriately “paddable” NP-hard sets, and the stronger claim we have stated here follows from the 1-truth-table special case of the result of Ogiwara and Watanabe [34] that if any NP-bounded-truth-table-hard set is sparse then $P = NP$). And so such extremely good heuristic algorithms almost certainly do not exist. But what about less ambitious hopes? Can any polynomial-time algorithm agree with any NP-hard set except for at most $n^{\log^{\mathcal{O}(1)} n}$ strings at each length, i.e., the symmetric difference between the set accepted by the algorithm and the NP-hard set has density $n^{\log^{\mathcal{O}(1)} n}$ (equivalently, $2^{\log^{\mathcal{O}(1)} n}$)? The answer is “no,” unless shocking complexity-theoretic collapses occur. To see this, we simply need to note that if such an algorithm existed, then the NP-hard set would polynomial-time 1-truth-table reduce (indeed, it would even reduce by a polynomial-time 1-truth-table reduction that was in addition restricted to a single truth-table, namely, the parity truth-table) to a set of density $n^{\log^{\mathcal{O}(1)} n}$. However, it is known ([30, 6]) that if any NP-hard set even polynomial-time $\mathcal{O}(1)$ -truth-table reduces (i.e., polynomial-time bounded-truth-table reduces) to a set of density $n^{\log^{\mathcal{O}(1)} n}$, then (i) all NP sets can be deterministically solved in time $n^{\log^{\mathcal{O}(1)} n}$, and (ii) $EXP = NEXP$ (where $EXP = \bigcup_{\text{polynomials } p} DTIME[2^{p(n)}]$ and $NEXP = \bigcup_{\text{polynomials } p} NTIME[2^{p(n)}]$), i.e., deterministic and nondeterministic exponential time coincide. Both of these consequences are broadly believed not to hold. (Various additional unlikely collapse consequences follow for the case where we are speaking not merely of NP-hard sets but in fact of NP-complete sets [6], such as the few hard problems discussed in this paper.) So polynomial-time heuristic algorithms are extremely unlikely to be able to come within $n^{\log^{\mathcal{O}(1)} n}$ errors-per-length of any NP-hard set.³

7 Conclusions and Open Directions

Motivated by the fact that real-world electorates are unlikely to be flawlessly single-peaked, we have studied the complexity of manipulative actions on nearly single-peaked electorates. We observed a wide range of behavior. Often, a modest amount of non-single-peaked behavior is not enough to obliterate an existing polynomial-time claim. We find this the most important theme of this paper—its “take-home message.” So if one feels that previous polynomial-time manipulative-action algorithms for single-peaked electorates are suspect since real-world electorates tend not to be truly single-peaked but rather nearly single-peaked, our results of this sort should reassure one on this point—although they are but a first step, as the paragraph after this one will explain. Yet we also found that sometimes allowing even one deviant voter is enough to raise the complexity from P to NP-hardness, and sometimes allowing any number of deviant voters has no effect at all on the complexity. We also saw cases where frequency of mavericity extracted a price in terms of amount of nondeterminism used. We feel this is a connection that should be further explored, and regarding Corollary 4.3, we particularly commend to the reader’s attention the issue of proving completeness for—not

³One may wonder how close to NP-hard sets *can* polynomial-time heuristic algorithms easily come? By using easy “padding” constructions, it is not hard to see that for each positive integer k it holds that every NP-hard set is polynomial-time equivalent to an NP-hard set that differs from the empty set—which itself certainly is in P —on at most $2^{n^{1/k}}$ of the 2^n length- n strings, i.e., they agree on all but an exponentially vanishing portion of the domain; see the construction in Footnote 10 of [13] for how to do this, except with the “2” there being changed to $k + 1$ (see also the comments/discussion in Appendix C of [12]). The best way to interpret this is not to just say that all NP-hard sets are akin to easy sets, but rather to realize that frequency of easiness is not a robust concept with respect to polynomial-time reductions, and neither is average-time complexity computed simply by averaging. Indeed, this type of effect is why average-case complexity theory is defined in ways far more subtle and complex than just taking a straightforward averaging of running times [33].

merely membership in—the levels of the beta hierarchy. We conjecture that completeness holds.

One might wish to study other notions of closeness to single-peakedness and, in particular, one might want to combine our notions. Indeed, in real human elections, there probably are both mavericks and swooners, and so our models are but a first step. In addition, the types of nearness that appear in different human contexts may differ from each other, and from the types of nearness that appear in computer multiagent systems contexts. Models of human/multiagent-system behavior, and empirical study of actual occurring vote sets, may help identify the most appropriate notions of nearness for a given setting.

Our control work studies just one type of control-attack at a time. We suspect that many of our polynomial-time results could be extended to handle multiple types of attacks simultaneously, as has recently been explored (without single-peakedness constraints) by Faliszewski et al. [17], and we mentioned in passing one result for which we have already shown this.

Acknowledgments We thank the anonymous referees for their comments. Piotr Faliszewski was supported in part by AGH University of Technology grant 11.11.120.865, by Polish Ministry of Science and Higher Education grant N-N206-378637, and by the Foundation for Polish Science’s Homing/Powroty program. Edith Hemaspaandra was supported in part by grant NSF-IIS-0713061 and a Friedrich Wilhelm Bessel Research Award. Lane A. Hemaspaandra was supported in part by grants NSF-CCF-0915792 and arc-dp110101792, and a Friedrich Wilhelm Bessel Research Award.

References

- [1] J. Bartholdi, III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
- [2] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.
- [3] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.
- [4] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proc. of AAAI-10*, pages 715–722, July 2010.
- [5] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. Technical Report TR-955, Department of Computer Science, University of Rochester, Rochester, NY, April 2010.
- [6] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles, and the exponential hierarchy. In *Proc. of FSTTCS-92*, pages 116–127. Springer-Verlag *Lecture Notes in Computer Science* #652, December 1992.
- [7] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *JAIR*, 35:161–191, 2009.
- [8] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

- [9] J. Díaz and J. Torán. Classes of bounded nondeterminism. *Mathematical Systems Theory*, 23(1):21–32, 1990.
- [10] E. Elkind, P. Faliszewski, and A. Slinko. On the role of distances in defining voting rules. In *Proc. of AAMAS-10*, pages 375–382, 2010.
- [11] G. Erdélyi and J. Rothe. Control complexity in fallback voting. In *Proc. of 16th Australasian Theory Symposium*, pages 39–48, January 2010.
- [12] G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. Frequency of correctness versus average-case polynomial time and generalized juntas. Technical Report TR-934, Department of Computer Science, University of Rochester, Rochester, NY, June 2008.
- [13] G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. Generalized juntas and NP-hard sets. *Theoretical Computer Science*, 410(38–40):3995–4000, 2009.
- [14] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.
- [15] G. Erdélyi, L. Piras, and J. Rothe. Bucklin voting is broadly resistant to control. Technical Report arXiv:1005.4115 [cs.GT], arXiv.org, May 2010.
- [16] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *Proc. of ECAI-08*, pages 366–370, July 2008.
- [17] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode attacks on elections. In *Proc. of IJCAI-09*, pages 128–133, July 2009.
- [18] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *JAIR*, 35:485–532, 2009.
- [19] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In *Fundamental Problems in Computing*, pages 375–406. Springer, 2009.
- [20] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. In *Proc. of TARK-09*, pages 118–127, July 2009. Full version appears as [23].
- [21] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *JAIR*, 35:275–341, 2009.
- [22] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.
- [23] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209:89–107, 2011.

- [24] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *Proc. of FOCS-08*, pages 243–249, October 2008.
- [25] S. Gailmard, J. Patty, and E. Penn. Arrow’s theorem on single-peaked domains. In E. Aragonés, C. Beviá, H. Llavador, and N. Schofield, editors, *The Political Economy of Democracy*, pages 335–342. Fundación BBVA, 2009.
- [26] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [27] J. Goldsmith, M. Levy, and M. Mundhenk. Limited nondeterminism. *SIGACT News*, 27(2):20–29, 1996.
- [28] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.
- [29] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.
- [30] S. Homer and L. Longpré. On reductions of NP sets to sparse sets. *Journal of Computer and System Sciences*, 48(2):324–336, 1994.
- [31] C. Kintala and P. Fisher. Refining nondeterminism in relativized polynomial-time bounded computations. *SIAM Journal on Computing*, 9(1):46–53, 1980.
- [32] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.
- [33] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.
- [34] M. Ogiwara and O. Watanabe. On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. *SIAM Journal on Computing*, 20(3):471–483, June 1991.
- [35] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *JAIR*, 28:157–181, 2007.
- [36] U. Schöning. Complete sets and closeness to complexity classes. *Mathematical Systems Theory*, 19(1):29–42, 1986.
- [37] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proc. of AAAI-07*, pages 3–8, July 2007.
- [38] T. Walsh. Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In *Proc. of IJCAI-09*, pages 324–329, July 2009.

A Formal Definitions

In this section we provide the missing formal definitions of the problems that we study (variants of manipulation, control, and bribery) and formal definitions of the problems we reduce from in our hardness proofs.

Definition A.1 ([8]). *Let R be an election system. In the CCWM problem for R we are given a set of candidates C , a preferred candidate $p \in C$, a collection of nonmanipulative voters S (each vote consists of a preference order and a positive integer, the weight of the vote), and a collection T of n manipulators, each specified by its positive integer weight. We ask if it is possible to set the preference orders of the manipulators in such a way that p is a winner of the resulting R election $(C, S \cup T)$.*

The following control notions are due to the seminal paper of Bartholdi et al. [3], except the notion below of CCAC follows Faliszewski et al. [21] in employing a bound, K , to make it better match the other control types.

Definition A.2 ([3]). *Let R be an election system.*

- (a) *In the CCAC problem for R we are given two disjoint sets of candidates, C and A , a collection V of votes over $C \cup A$, a candidate $p \in C$, and a nonnegative integer K . We ask if there is a set $A' \subseteq A$ such that (a) $\|A'\| \leq K$, and (b) p is a winner of R election $(C \cup A', V)$.*
- (b) *In the CCDC problem for R we are given an election (C, V) , a candidate $p \in C$, and a nonnegative integer K . We ask if there is a set $C' \subseteq C$ such that (a) $\|C'\| \leq K$, (b) $p \notin C'$, and (c) p is a winner of R election $(C - C', V)$.*
- (c) *In the CCAV problem for R we are given a set of candidates C , two collections of voters, V and W , over C , a candidate $p \in C$, and a nonnegative integer K . We ask if there is a subcollection $W' \subseteq W$ such that (a) $\|W'\| \leq K$, and (b) p is a winner of R election $(C, V \cup W')$.*
- (d) *In the CCDV problem for R we are given an election (C, V) , a candidate $p \in C$, and a nonnegative integer K . We ask if there is a collection V' of voters that can be obtained from V by deleting at most K voters such that p is a winner of R election (C, V') .*

The bribery notions below are due to Faliszewski et al. [18], except the notion below of negative and strongnegative bribery for approval voting are due to Brandt et al. [4].

Definition A.3 ([18, 4]). *Let R be an election system. In the weighted-\$bribery problem for R we are given an election (C, V) , where each vote consists of the voter's preferences (as appropriate for the election system, e.g., an approval vector for approval voting and a preference order for plurality) and two integers (this vote's positive integer weight and this vote's nonnegative integer price), a candidate $p \in C$, and a nonnegative integer K (the allowed budget). We ask if there is a subcollection of votes, whose total price does not exceed K , such that it is possible to ensure that p is an R -winner of the election by modifying the preferences of these votes.*

The problems (a) weighted-bribery, (b) \$bribery, and (c) bribery for R are variants of weighted-\$bribery for R where, respectively: (a) no prices are specified and each vote is treated as having unit cost, (b) no weights are specified, and each vote is treated as having unit weight, and (c) no prices or weights are specified, and each vote is treated as having unit price and unit weight.

For plurality, “negative” bribery means no bribed voter can have p as the most preferred candidate in his/her preference order.

For approval voting, “negative” bribery means a bribe cannot change someone from disapproving of p to approving of p , and “strongnegative” bribery means every bribed voter must end up disapproving of p .

Definition A.4 (see, for example, Garey and Johnson [26]). A *PARTITION* instance I is a set of $\{k_1, \dots, k_n\}$ of n distinct positive integers that sums to $2K$. I is a yes instance if there exists a subset of $\{k_1, \dots, k_n\}$ that sums to K and it is a no instance otherwise. An *X3C* instance $I = (B, \mathcal{S})$ consists of a base set $B = \{b_1, \dots, b_{3k}\}$ and a family $\mathcal{S} = \{S_1, \dots, S_n\}$ of three-element subsets of B . I is a yes instance if it is possible to pick exactly k sets from \mathcal{S} so that their union is B and it is a no instance otherwise.

Definition A.5. For score-based election systems (e.g., plurality, approval, scoring protocols), we write $\text{score}_{(C,V)}(c)$ to denote the score of candidate c in election (C, V) ; naturally we require that $c \in C$. The particular election system that we use will always be clear from context.

B Proofs from Section 3

Theorem 3.1. For each $\alpha_1 \geq \alpha_2 > \alpha_3$, CCWM for $(\alpha_1, \alpha_2, \alpha_3)$ elections over 1-maverick-SP societies is NP-complete.

Proof of Theorem 3.1. Without loss of generality, we assume that $\alpha_3 = 0$. We will reduce from PARTITION. Given a set $\{k_1, \dots, k_n\}$ of n distinct positive integers that sums to $2K$, define the following instance of CCWM. Let $C = \{p, a, b\}$, let society’s order be $aLpLb$, let S consist of one voter with preference order $a > b > p$ (note that this voter is the maverick) with weight $(2\alpha_1 - \alpha_2)\alpha_1 K$, and one voter with preference order $b > p > a$ with weight $(2\alpha_1 - \alpha_2)(\alpha_1 - \alpha_2)K$. (Technically, weights need to be positive, but if $\alpha_1 = \alpha_2$ we can get the same effect by letting S consist of just the maverick.) Note that $\text{score}_{(C,S)}(a) = \text{score}_{(C,S)}(b) = (2\alpha_1^3 - \alpha_1^2\alpha_2)K$ and that $\text{score}_{(C,S)}(p) = (2\alpha_1 - \alpha_2)(\alpha_1 - \alpha_2)\alpha_2 K = (2\alpha_1^2\alpha_2 - 3\alpha_1\alpha_2^2 + \alpha_2^3)K$. Let T consist of n manipulators with weights $(\alpha_1^2 - \alpha_1\alpha_2 + \alpha_2^2)k_1, \dots, (\alpha_1^2 - \alpha_1\alpha_2 + \alpha_2^2)k_n$.

If there is a subset of k_1, \dots, k_n that sums to K , then we let all manipulators in T whose weight divided by $(\alpha_1^2 - \alpha_1\alpha_2 + \alpha_2^2)$ is in this subset vote $p > a > b$, and all manipulators in T whose weight divided by $(\alpha_1^2 - \alpha_1\alpha_2 + \alpha_2^2)$ is not in this subset vote $p > b > a$. It is immediate that $\text{score}_{(C,S \cup T)}(a) = \text{score}_{(C,S \cup T)}(b) = \text{score}_{(C,S)}(a) + (\alpha_1^2\alpha_2 - \alpha_1\alpha_2^2 + \alpha_2^3)K = (2\alpha_1^3 - \alpha_1\alpha_2^2 + \alpha_2^3)K$ and that $\text{score}_{(C,S \cup T)}(p) = \text{score}_{(C,S)}(p) + (2\alpha_1^3 - 2\alpha_1^2\alpha_2 + 2\alpha_1\alpha_2^2)K = (2\alpha_1^3 - \alpha_1\alpha_2^2 + \alpha_2^3)K$. It follows that all candidates are tied, and thus all candidates are winners.

For the converse, suppose the manipulators vote so that p becomes a winner. It is easy to see that we can assume that all manipulators rank p first. From the calculations above, it is also easy to see that it is always the case that $2\text{score}_{(C,S \cup T)}(p) \leq \text{score}_{(C,S \cup T)}(a) + \text{score}_{(C,S \cup T)}(b)$. In order for p to be a winner, we thus certainly need the scores of a and b to be equal. This implies that $\text{score}_{(C,T)}(a) = \text{score}_{(C,T)}(b)$. But then the weights of the manipulators voting $p > a > b$ sum to $(\alpha_1^2 - \alpha_1\alpha_2 + \alpha_2^2)K$. \square

Theorem 3.2. For each $k \geq 0$ and $m \geq k + 3$, CCWM for m -candidate veto elections over k -maverick-SP societies is in P.

Proof. Let $k \geq 0$ and $m \geq k + 3$, and let society’s order be L . Let c_ℓ be the leftmost candidate in L and let c_r be the rightmost candidate in L . In an m -candidate veto election, the $m - 2$ candidates in $C - \{c_\ell, c_r\}$ are never vetoed by the nonmavericks. Every maverick vetoes at most one of these $m - 2$ candidates. Since $k < m - 2$, in a k -maverick m -candidate veto election, there exists a candidate that is never vetoed. So, given

an instance of CCWM for m -candidate veto elections over k -maverick-SP societies, p can be made a winner if and only if p is never vetoed by the nonmanipulators. \square

Theorem 3.3. *For each $k \geq 0$ and $m \geq 3$ such that $m \leq k + 2$, CCWM for m -candidate veto elections over k -maverick-SP societies is NP-complete.*

Proof. We will again reduce from PARTITION. Given a set $\{k_1, \dots, k_n\}$ of n distinct positive integers that sums to $2K$, define the following instance of CCWM. Let $C = \{p, a, b, c_1, \dots, c_{m-3}\}$, let society's order be $aLpLc_1L \dots Lc_{m-3}Lb$, let S consist of $m - 2$ voters each of weight K . For every candidate c in $C - \{a, b\}$ there is a voter in S that ranks c last. Note that all voters in S are mavericks. This is allowed, since $m - 2 \leq k$. Let T consist of n manipulators with weights k_1, \dots, k_n .

If there is a subset of k_1, \dots, k_n that sums to K , then we let all manipulators in T whose weight is in this subset vote $a > p > c_1 > \dots > c_{m-3} > b$, and all manipulators in T whose weight is not in this subset vote $b > c_{m-3} > \dots > c_1 > p > a$. It is immediate that all candidates in election $(C, S \cup T)$ are tied and so p is a winner.

For the converse, suppose the manipulators can vote so that p becomes a winner. Note that p needs to gain at least K points over a and over b in T . Clearly, the only way this can happen is if $\text{score}_{(C,T)}(a) = \text{score}_{(C,T)}(b) = K$. But then the weights of the voters in T who rank a last add to K . \square

Theorem 3.5. *For each $m \geq 5$, CCWM for m -candidate veto elections in swoon-SP societies is in P. For $m \in \{3, 4\}$, this problem is NP-complete.*

Proof. First suppose that $m \geq 5$. Let L be society's order. Let c be a candidate such that there are at least two candidates to the left of c in L and there are at least two candidates to the right of c in L . In an m -candidate veto election in a swoon-SP society, c will never be vetoed. Given a CCWM instance for m -candidate veto elections in swoon-SP societies, p can be made a winner if and only if p is never vetoed by the nonmanipulators.

Now consider the case that $m = 4$. We will reduce from PARTITION. Given a set $\{k_1, \dots, k_n\}$ of n distinct positive integers that sums to $2K$, define the following instance of CCWM. Let $C = \{p, a, b, c\}$, let society's order be $aLpLbLc$, let S consist of two voters, each with weight K . One voter in S votes $a > c > b > p$ and one voter votes $c > a > p > b$. Let T consist of n manipulators with weights k_1, \dots, k_n .

If there is a subset of k_1, \dots, k_n that sums to K , then we let all manipulators in T whose weight is in this subset veto a and all manipulators in T whose weight is not in this subset veto c . It is immediate that all candidates in election $(C, S \cup T)$ are tied and so p is a winner.

For the converse, suppose the manipulators can vote so that p becomes a winner. Note that p needs to gain at least K points over a and over c in T . Clearly, the only way this can happen is if $\text{score}_{(C,T)}(a) = \text{score}_{(C,T)}(c) = K$. But then the weights of the voters in T who veto a add to K .

A very similar proof can be used to show the statement for $m = 3$. However, the statement for $m = 3$ also follows immediately from the fact that CCWM for 3-candidate veto elections is NP-complete and the observation below that every 3-candidate election is a swoon-SP election. \square

Observation 3.6. *Every 3-candidate election is a swoon-SP election and a Dodgson₁-SP election and so all complexity results for 3-candidate elections in the general case also hold for swoon-SP elections and Dodgson₁-SP elections.*

Proof. Since every 2-candidate vote is single-peaked, it follows immediately that every 3-candidate election is a swoon-SP election. For the Dodgson₁-SP case, suppose society's order is $aLbLc$. The only votes that

are not single-peaked are $a > c > b$ and $c > a > b$. But note that both of these votes are one adjacent swap away from being single-peaked, by swapping the last two candidates. \square

Theorem 3.7. *For each $m \geq 5$, CCWM for m -candidate veto elections in Dodgson_1 -SP societies is in P. For $m \in \{3, 4\}$, this problem is NP-complete.*

Proof. The $m \geq 5$ case follows using the same proof as the $m \geq 5$ case of Theorem 3.5. The $m = 4$ case follows using the same proof as the $m = 4$ case of Theorem 3.5 except that, in order for the votes to be within one adjacent swap of being consistent with societal order, the two voters in S now vote $c > b > a > p$ and $a > p > c > b$. The $m = 3$ case follows from Observation 3.6. \square

Theorem 3.8. *For each $\alpha_1 \geq \alpha_2 > \alpha_3$, CCWM for $(\alpha_1, \alpha_2, \alpha_3)$ elections over single-caved societies is NP-complete if $(\alpha_1 - \alpha_3) \leq 2(\alpha_2 - \alpha_3)$ and otherwise is in P.*

Proof. Without loss of generality, assume that $\alpha_3 = 0$. We first consider the case that $\alpha_1 > 2\alpha_2$. Let (C, V) be an $(\alpha_1, \alpha_2, \alpha_3)$ election over single-caved societies, let W be the total weight of V , and let L be society's order. Consider the middle candidate in L . This candidate can never be ranked first, and so its score will be at most $\alpha_2 W$, and the sum of the scores of the other two candidates will be at least $\alpha_1 W$. Since $\alpha_1 > 2\alpha_2$, it follows that the middle candidate will never be a winner if $W > 0$. Given an instance of CCWM for $(\alpha_1, \alpha_2, 0)$ elections over single-caved societies, p can be made a winner if and only if (1) p is the middle candidate in L and $W = 0$, or (2) p is not the middle candidate in L and p is a winner if all manipulators rank p first, then the middle candidate, and then the last candidate. All this is easy to check in polynomial time.

Now consider the case that $\alpha_1 \leq 2\alpha_2$. We will show that in this case PARTITION can be reduced to CCWM for $(\alpha_1, \alpha_2, 0)$ elections over single-caved societies. Given a set $\{k_1, \dots, k_n\}$ of n distinct positive integers that sums to $2K$, define the following instance of CCWM. Let $C = \{p, a, b\}$ and let society's order be $aLpLb$, and let S consist of two voters, each with weight $(2\alpha_2 - \alpha_1)K$. One voter in S votes $a > b > p$ and one voter votes $b > a > p$. (Technically, weights need to be positive, but if $\alpha_1 = 2\alpha_2$ we can get the same effect by letting $S = \emptyset$.) Let T consist of n manipulators with weights $(\alpha_1 + \alpha_2)k_1, \dots, (\alpha_1 + \alpha_2)k_n$.

If there exists a subset of $\{k_1, \dots, k_n\}$ that sums to K , we let the manipulators whose weight divided by $(\alpha_1 + \alpha_2)$ is in this subset vote $a > p > b$ and the manipulators whose weight divided by $(\alpha_1 + \alpha_2)$ is not in this subset vote $b > p > a$. It is easy to see that $\text{score}_{(C, S \cup T)}(a) = \text{score}_{(C, S \cup T)}(b) = (2\alpha_2 - \alpha_1)(\alpha_1 + \alpha_2)K + \alpha_1(\alpha_1 + \alpha_2)K = 2\alpha_2(\alpha_1 + \alpha_2)K = \text{score}_{(C, S \cup T)}(p)$, and so p is a winner.

For the converse, suppose p can be made a winner. Since p is the middle candidate in L , p can not be ranked first. Without loss of generality we can assume that the voters in T vote $a > p > b$ or $b > p > a$. It follows that $\text{score}_{(C, S \cup T)}(p) = 2\alpha_2(\alpha_1 + \alpha_2)K$. Since $\text{score}_{(C, S \cup T)}(a) + \text{score}_{(C, S \cup T)}(b) = 4\alpha_2(\alpha_1 + \alpha_2)K$, by the argument above, the only way p can be a winner is if a and b tie in T . But then the weights of the manipulators voting $a > p > b$ sum to $(\alpha_1 + \alpha_2)K$. \square

C Proofs from Section 4

In the following subsections we provide the missing proofs from Section 4.

C.1 Proofs of Theorems 4.1, 4.4, and 4.6

Theorem 4.1. *CCAV and CCDV for approval elections over log-maverick-SP societies are each in P. For CCAV, the complexity remains in P even for the case where no limit is imposed on the number of mavericks in*

the initial voter set, and the number of mavericks in the set of potential additional voters is logarithmically bounded (in the overall problem input size).

Proof. Consider the case of CCAV. We will handle directly the stronger case mentioned in the theorem, namely the one with no limit on the number of mavericks in the initial voter set. There of course will be a $\mathcal{O}(\log(\text{ProblemInputSize}))$ limit on the number of mavericks in the set of voters to potentially add. Let that (easy, nondecreasing) upper-bounding function be called f .

So, suppose we are given an input instance of this problem. Let K , which is part of the input, be the limit on the number of voters we are allowed to add. We start by doing the obvious syntactic checks, and we also check that the number of voters in the additional voter set who are not consistent with the input societal order is at most $f(\text{ProblemInputSize})$. If any of these checks fail, we reject.

Now, we will show how to build a polynomially long list of instances of the CCAV problem over single-peaked elections such that our control goal is possible to achieve *if and only if* one or more of those control problems has a goal that can be achieved. (That is, we will implicitly give a polynomial-time disjunctive truth-table reduction to the single-peaked case.)

Our construction is as follows. For each choice of which mavericks from the additional voter set to add to our election, we will generate at most one instance of a single-peaked control question. Since there are at most logarithmically many such mavericks, and the number of cases we have to look at is the cardinality of the powerset of the number of mavericks among the additional voters, the number of instances we generate is polynomially bounded.

For each choice A of which mavericks from among the additional voter set to add to the main election, we generate at most one instance as follows. If $\|A\| > K$, we will generate no instance, as that choice is trying to add illegally many additional voters. Otherwise, we generate a single-peaked election instance as follows. Move the elements of A from the additional voter set to the main election. Remove all remaining mavericks from the additional voter set. Demaverickify our election as follows: For each maverick voter v , for each candidate c that v approves, add a new voter who approves of only c (and so certainly is consistent with the single-peaked societal order). Then remove all the maverick voters. Note that this demaverickification process does not change the approval counts of the election and does ensure that the electorate is single-peaked. The entire demaverickification process does not increase the problem's size by more than a polynomial, since no voter is replaced by more than $\|C\| - 1$ voters. Replace K by $K - \|A\|$. The resulting instance is the instance that this choice of A adds to our collection of instances.

So, we have created a polynomial-length list of (polynomial-sized) instances of the single-peaked CCAV problem. It is easy to see that our control goal can be achieved exactly if for at least one of these instances the control goal can be achieved. Briefly put, that is because our problem has a successful control action (after passing the initial maverick-cardinality-limit check) exactly if there is some appropriate-sized subset of additional voters that we can add to make the favored candidate become a winner. Our above process tries every legal set of choices for which mavericks from the additional voter set might be the mavericks in the added set. And the instance it generates, based on that choice, will have a successful control action precisely if what remains of our initial K bound, after we remove the cardinality of the added mavericks, is such that there is some number of nonmaverick additional voters who can be added to achieve the desired victory for p . In addition, the instance generated is a single-peaked society, and the transformation we used to make it single-peaked doesn't in anyway affect the answer to the created instance, since the demaverickification occurred only on voters that were (at that point, although some had not started there) in our main voter set, and the only affect that set has on the single-peaked CCAV control question is the approval totals of each candidate, and our demaverickification did not alter those totals.

Our polynomial-length list of instances is composed just of instances of the CCAV problem for approval

elections over single-peaked electorates. That problem has a polynomial-time algorithm [23]. And so we run that algorithm on each of the polynomially many instances, and if any finds a successful control action, our original problem has a successful control action, and if not our original problem does not. Thus, our proof of the CCAV case is complete.

However, a final comment is needed, since we wish to not only give a yes/no answer, but to in fact find what control action to take, when one is possible. (Doing so goes beyond what the theorem promises, but we in general will try to give algorithms that not only give yes/no answers but also that at least implicitly make available the actual successful actions for the yes instances.) Formally speaking, disjunctive truth-table reductions are about languages, rather than about solutions. Nonetheless, from our construction it is immediately clear how a successful control action for any problem on the list—and the polynomial-time algorithm of Faliszewski et al. [23] in fact gives not merely a yes/no answer but in fact finds a successful control action when one exists—specifies a successful control action for our nearly-single-peaked original problem.

The CCDV case might at first seem to be almost completely analogous, except that in that case, there is no separate pool of additional voters, and the logarithmic bound applies to the entire set of initial voters. However, the reduction approach we took above for CCAV at first seems not to work here. The reason is that for the CCAV case, the mavericks we added could be demaverickified in a way that didn’t interfere with the call to the single-peaked case of the CCAV approval voting algorithm, *and the mavericks we decided not to add could (for the instance being generated) be deleted*. In contrast, for the CCDV case, whichever mavericks we don’t delete remain very much a part of the election—and are indeed part of the instance we would like to generate of a case of a call to CCDV. But that means the generated case may not be single-peaked, as we would like it to be.

We can work around this obstacle, by noting that the algorithm given in Faliszewski et al. [23] for the single-peaked CCDV approval-voting case in fact does a bit more than is claimed there. It is easy to see, looking at that paper, that it in effect gives a polynomial-time algorithm for the following problem: Given an instance of CCDV, and a societal ordering, and given that in the instance’s voter set each voter has an extra bit specifying whether the voter is deletable or is not deletable, and given that every voter that is specified as being deletable must be consistent with the societal ordering (but voters specified as being not deletable are not required to be consistent with the societal ordering—they may be mavericks), is there a set of at most K (K being part of the input) deletable voters such that if we delete them our preferred candidate p is a winner? The fact that the paper implicitly gives such an algorithm is clear from that paper, since regarding the “deleting voters” actions described on its page 96 we can choose to allow them only on the deletable voters, and the Faliszewski et al. [23] algorithm’s correctness in our case hinges (assuming that nondeletable voters are indeed nondeletable) just on the fact that the deletable voters all respect the societal ordering. (Once we allow a deletable/nondeletable flag, we could in fact demaverickify all the remaining mavericks, and then flag all the 1-approval-each voters added by that demaverickification as being nondeletable, but there is no need to do any of that. Doing it requires the deletable/nondeletable flag, and as just noted, if one has that flag, one can outright tolerate (nondeletable) mavericks.)

So, we have noted a polynomial-time algorithm that, while not stated as their theorem, is a corollary to their theorem’s proof—i.e., the proof of the CCDV result that in [23] appears on that paper’s page 93. With this in hand, we can handle our nearly-single-peaked CCDV case using the same basic approach we used for CCAV, as naturally modified for the CCDV case. In particular, we again polynomial-time disjunctive truth-table reduce to a problem known to be in polynomial time—in this case, CCDV for approval voting over single-peaked societies, with a deletable/nondeletable flag for each voter, and with all deletable voters having to be nonmavericks, which was argued above to be in polynomial time. Our reduction is that (after

checking that the input election is syntactically correct and does not have illegally many mavericks) for each subset of the mavericks that is of cardinality at most K (the input bound on the number of voters to delete), we delete those K mavericks, then we decrement K by the cardinality of the subset, then we mark all the nonmaverick voters as deletable, and mark each remaining maverick as nondeletable. This approach works for essentially the same reason as the CCAV case, and as in that case, we can get not merely a yes/no answer, but can even for the yes cases produce a successful control action. \square

Theorem 4.4. *CCAV and CCDV for Condorcet elections over $f(\cdot)$ -maverick-SP societies are each in $\text{NDET-TIME}[f(\text{ProblemInputSize}), \text{poly}]$. For CCAV, the complexity remains in $\text{NDET-TIME}[f(\text{ProblemInputSize}), \text{poly}]$ even for the case where no limit is imposed on the number of mavericks in the initial voter set, and the number of mavericks in the set of potential additional voters is $f(\cdot)$ -bounded (in the overall problem input size).*

Proof. Let us handle the CCAV case first. We assume we are in the more general setting where there is no limit on the number of mavericks in the initial voter set. Let (C, V, W, p, K) be our input instance of CCAV for Condorcet and let L be the societal axis. We assume that the candidate set C is of the form $C = \{b_{m'}, \dots, b_1, p, c_1, \dots, c_{m''}\}$ and $b_{m'} L \dots L b_1 L p L c_1 L \dots L c_{m''}$ holds. We partition the voters in W into four groups, W_ℓ , W_r , W_p , and W_m :

1. W_ℓ contains those voters from W who are not mavericks and whose most preferred candidate c is such that $c L p$ (intuitively, these are the voters whose top choice is “to the left of p ”).
2. W_r contains those voters from W who are not mavericks and whose most preferred candidate c is such that $p L c$ (intuitively, these are the voters whose top choice is “to the right of p ”).
3. W_p contains those voters from W whose most preferred candidate is p .
4. W_m contains the remaining voters from W (i.e., W_m contains those mavericks who do not rank p first; note that there are at most $f(\text{ProblemInputSize})$ voters in W_m).

Voters in W_ℓ (in W_r) have an interesting structure of their preference orders. For each $v \in W_\ell$ (each $v \in W_r$), due to his or her single-peakedness, there is a positive integer i such that v prefers each candidate in $\{b_1, \dots, b_i\}$ to p to each of the remaining candidates (v prefers each candidate in $\{c_1, \dots, c_i\}$ to p to each of the remaining candidates). Thus, we can conveniently sort the voters in W_ℓ (the voters in W_r) in increasing order of the cardinalities of the sets of candidates they prefer to p .

Our (nondeterministic) algorithm works as follows. First, we add $\max(\|W_p\|, K)$ voters from W_p . If that makes p a Condorcet winner, we accept. Otherwise, we set $K' = K - \max(\|W_p\|, K)$. If $K' = 0$, we reject. Next, using (at most) $f(\text{ProblemInputSize})$ nondeterministic binary decisions, for each voter v in W_m we decide whether to add v to the election or not. Let M be the number of voters we add in this process. If $M > K'$ then we reject (on this computation path) and otherwise we set $K'' = K' - M$. Then, we execute the following algorithm:

1. For each two nonnegative integers K_ℓ and K_r such that $K_\ell + K_r \leq K''$, execute the following steps.
 - (a) Add K_ℓ voters from W_ℓ to the election (in the order described one paragraph above).
 - (b) Add K_r voters from W_r to the election (in the order described one paragraph above).
 - (c) Check if p is the Condorcet winner of the resulting election. If so, accept. Otherwise, undo the adding of the voters from the two preceding steps.

2. If we have not accepted until this point, reject on this computation path.

It is easy to verify that this algorithm indeed runs in polynomial time (given access to $f(\text{ProblemInputSize})$ nondeterministic steps). Its correctness follows naturally from the observations regarding the preference orders of voters in W_ℓ and W_r (it is clear that, given K_ℓ and K_r , our algorithm adds K_ℓ voters from W_ℓ in an optimal way and adds K_r voters from W_r in an optimal way).

Let us now turn to the case of CCDV. The algorithm is very similar. Let (C, V, p, K) be our input instance. We assume that the candidate set C is of the form $C = \{b_{m'}, \dots, b_1, p, c_1, \dots, c_{m''}\}$ and $b_{m'} L \dots L b_1 L p L c_1 L \dots L c_{m''}$ holds. We partition the voters in V into four groups, V_ℓ , V_r , V_p , and V_m :

1. V_ℓ contains those voters from V who are not mavericks and whose most preferred candidate c is such that $c L p$ (intuitively, these are the voters whose top choice is “to the left of p ”).
2. V_r contains those voters from V who are not mavericks and whose most preferred candidate c is such that $p L c$ (intuitively, these are the voters whose top choice is “to the right of p ”).
3. V_p contains those voters from V whose most preferred candidate is p .
4. V_m contains the remaining voters from V (i.e., V_m contains those mavericks who do not rank p first; note that there are at most $f(\text{ProblemInputSize})$ voters in V_m).

For each $v \in V_\ell$ (each $v \in V_r$), due to his or her single-peakedness, there is a positive integer i such that v prefers each candidate in $\{b_1, \dots, b_i\}$ to p to each of the remaining candidates (v prefers each candidate in $\{c_1, \dots, c_i\}$ to p to each of the remaining candidates). Thus, we can conveniently sort the voters in W_ℓ (the voters in W_r) in decreasing order of the cardinalities of the sets of candidates they prefer to p .

It is clear that we should never delete voters from V_p . Our (nondeterministic) algorithm proceeds as follows. First, for each voter v in V_m we make a nondeterministic decision whether to delete v from the election or not. Let M be the number of voters we delete in this process. If $M > K$ then we reject on this computation path and otherwise we set $K' = K - M$. Next, we execute the following algorithm:

1. For each two nonnegative integers K_ℓ and K_r such that $K_\ell + K_r \leq K'$, execute the following steps:
 - (a) Delete K_ℓ voters from V_ℓ (in the order described one paragraph above).
 - (b) Delete K_r voters from V_r (in the order described one paragraph above).
 - (c) Check if p is the Condorcet winner of the resulting election. If so, accept. Otherwise, undo the deleting of the voters from the two preceding steps.
2. If we have not accepted so far, reject on this computation path.

Correctness and polynomial running time of the algorithm (given access to $f(\text{ProblemInputSize})$ nondeterministic steps) follow analogously as in the CCAV case. \square

Theorem 4.6. *For each k , CCAC and CCDC for plurality over k -maverick-SP societies are in P.*

Proof. The main idea of our proof is analogous to that of the proof of Theorem 4.1 but the details of demaverickification are different and, as a result, we can only handle a constant number of mavericks. We handle the CCAC case first.

Let $I = (C, A, V, p, K)$ be our input instance of CCAC for plurality and let L be the societal axis. Let k' be the number of mavericks in V ($k' \leq k$) and let $M = \{m_1, \dots, m_{k'}\}$ be the subcollection of V containing exactly these k' maverick voters. Our algorithm proceeds as follows:

1. For each vector $B = (b_1, \dots, b_{k'}) \in (C \cup A)^{k'}$ of candidates execute the following steps (intuitively, we intend to enforce that candidates $b_1, \dots, b_{k'}$ are top-ranked candidates of voters in M and that it is impossible to change the top-ranked candidates of voters in M by adding other candidates).
 - (a) If for any voter m_i , $1 \leq i \leq k'$, it holds that m_i prefers some candidate in $(C \cup \{b_j \mid 1 \leq j \leq k'\}) - \{b_i\}$ to b_i then drop this B and return to Step 1. (This condition guarantees that it is possible to ensure, via adding candidates from A , that for each i , $1 \leq i \leq k'$, voter m_i ranks candidate b_i first among the participating candidates.)
 - (b) Set $C' = C \cup \{b_j \mid 1 \leq j \leq k', b_j \in A\}$.
 - (c) Set $A' = (A - \{b_j \mid 1 \leq j \leq k', b_j \in A\}) - \{a \in A \mid \text{some voter } m_i, 1 \leq i \leq k', \text{ prefers } a \text{ to } b_i\}$.
 - (d) Set $K' = K - \|\{b_j \mid 1 \leq j \leq k', b_j \in A\}\|$. If $K' < 0$ then drop this B and return to Step 1.
 - (e) Form a voter collection V' that is identical to V except that we restrict voters' preferences to candidates in $C' \cup A'$ and for each voter m_i , $1 \leq i \leq k'$, we replace m_i 's preference order with an easily-computable preference order over $C' \cup A'$ that ranks b_i first and is single-peaked with respect to L .
 - (f) Using the polynomial-time algorithm of Faliszewski et al. [23], check if (C', A', V', p, K') is a yes instance of CCAC for single-peaked plurality elections with societal axis L . If so, accept.
2. If the algorithm has not accepted so far, reject.

It is easy to verify that the above algorithm indeed runs in polynomial time: There are exactly $\|C \cup A\|^{k'}$ choices of vector B to test and for each fixed B each step can clearly be performed in polynomial time. It remains to show that the algorithm is correct.

Let us assume that I is a yes instance. We will show that in this case the algorithm accepts. Let A'' be a subset of A such that $\|A''\| \leq K$ and p is a winner of election $E'' = (C \cup A'', V)$. Let $B'' = (b''_1, \dots, b''_{k'})$ be the vector of candidates from $C \cup A''$ such that for each i , $1 \leq i \leq k'$, in E'' voter m_i ranks b''_i first. We claim that our algorithm accepts at latest when it considers vector B'' . First, by our choice of B'' it is clear that in Step (1a) we do not drop B'' . Let C' , A' , K' , and V' be as computed by our algorithm for $B = B''$. By our choice of B'' , it is clear that $A'' - \{b''_i \mid 1 \leq i \leq k'\} \subseteq A'$. Thus, there is a set $A''' \subseteq A'$ such that $\|A'''\| \leq K'$ and p is a winner of election $(C' \cup A''', V)$. Further, since every voter m_i , $1 \leq i \leq k'$, prefers candidate b''_i to all other candidates in $C' \cup A'$, it holds that p is a winner of election $(C' \cup A''', V')$. That is, (C', A', V', p, K') is a yes instance.

Similarly, it is easy to see that the construction of instances (C', A', V', p, K') , and in particular the construction of V' in Step (1e), ensures that if the algorithm accepts then I is a yes instance. This completes the discussion of the CCAC case.

Let us now move on to the case of CCDC. As in the case of CCAC, we will, essentially, reduce the problem to the case where all voters are single-peaked. However, we will need the following more general variant of the CCDC problem.

Definition C.1. *Let R be an election system. In the CCDC with restricted deleting problem for R , we are given an election (C, V) , a candidate $p \in C$, a set $F \subseteq C$ such that $p \in F$, and a nonnegative integer K . We ask if there is a set $C' \subseteq C$ such that (a) $\|C'\| \leq K$, (b) $C' \cap F = \emptyset$, and (c) p is a winner of R election $(C - C', V)$.*

That is, in CCDC with restricted deleting we can specify which candidates are impossible to delete. The following result is a direct corollary to the proof of Faliszewski et al. [23] that CCDC for plurality is in P for single-peaked electorates.

Observation C.2 (Implicit in Faliszewski et al. [23]). *CCDC with restricted deleting is in P for plurality over single-peaked electorates.*

Let $I = (C, V, p, K)$ be our input instance of CCDC for plurality and let L be the societal axis. Let k' be the number of mavericks in V ($k' \leq k$) and let $M = \{m_1, \dots, m_{k'}\}$ be the subcollection of V that contains these k' maverick voters. Our algorithm works as follows:

1. For each vector $B = (b_1, \dots, b_{k'}) \in C^{k'}$ of candidates execute the following steps (intuitively, we intend to enforce that candidates $b_1, \dots, b_{k'}$ are top-ranked candidates of voters in M).
 - (a) If for any voter m_i , $1 \leq i \leq k'$, it holds that m_i prefers some candidate in $(C \cup \{b_j \mid 1 \leq j \leq k'\}) - \{b_i\}$ to b_i then drop this B and return to Step 1. (This condition guarantees that it is possible to ensure, via deleting voters, that for each i , $1 \leq i \leq k'$, voter m_i ranks candidate b_i first among the participating candidates.)
 - (b) Set $F' = \{b_i \mid 1 \leq i \leq k'\} \cup \{p\}$.
 - (c) Set $C' = C - \{c \in C \mid \text{there is an } i, 1 \leq i \leq k' \text{ such that } m_i \text{ prefers } c \text{ to } b_i\}$.
 - (d) Set $K' = K - \|\{c \in C \mid \text{there is an } i, 1 \leq i \leq k' \text{ such that } m_i \text{ prefers } c \text{ to } b_i\}\|$. If $K' < 0$ then drop this B and return to Step 1.
 - (e) Form a voter collection V' that is identical to V except that we restrict voters' preferences to candidates in C' and for each voter m_i , $1 \leq i \leq k'$, we replace m_i 's preference order with an easily-computable preference order over C' that ranks b_i first and is single-peaked with respect to L .
 - (f) Using Observation C.2, check if (C', V', p, F', K') is a yes instance of CCDC with restricted deleting for single-peaked plurality elections with societal axis L . If so, accept.
2. If the algorithm has not accepted so far, reject.

Using the same arguments as in the case of CCAC, we can see that this algorithm is both correct and runs in polynomial time. \square

C.2 Proofs of Theorems 4.7, 4.8, and 4.15

Theorem 4.7. *CCAC and CCDC for plurality elections over swoon-SP societies are NP-complete.*

Theorem 4.7 implies the classic result of Bartholdi et al. [3] that CCAC and CCDC are NP-complete for plurality elections. However, the proofs of Bartholdi, Tovey, and Trick cannot be used directly for swoon-SP societies.

Proof of Theorem 4.7. We first consider the case of CCAC. We easily note that CCAC for plurality over swoon-SP societies is in NP. It remains to show that it is NP-hard and we do so by giving a reduction from X3C. Let $I = (B, \mathcal{S})$ be our input X3C instance, where $B = \{b_1, \dots, b_{3k}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$. Without loss of generality, we assume that $k \geq 2$ and $n \geq 4$. For each $b_i \in B$, we set ℓ_i to be the number of sets in \mathcal{S} that contain b_i .

We construct an election $E = (C \cup A, V)$, where $C = B \cup \{p, d\}$ is the set of registered candidates, $A = \{a_1, \dots, a_n\}$ is the set of spoiler (unregistered) candidates, and V is a collection of votes. Each candidate a_i in A corresponds to a set S_i in \mathcal{S} . We assume that the societal axis L is $pLdLb_1L \cdots Lb_{3k}La_1L \cdots La_n$. (Our proof works for any easily computable axis.) Collection V contains the following $(6kn) + (n) + (2nk + k - n) + (2nk) + (\sum_{i=1}^{3k} (2nk + 2k - 2k\ell_i))$ votes; each of the five parenthesized terms in this expression corresponds to an item in the description of votes below. (For each vote we only specify up to two top-ranked candidates. Note that voters in swoon-SP societies can legally pick any two candidates to be ranked in the top two positions of their votes. This is so, because the top-ranked candidate can be chosen freely as the candidate to which the voter swoons, and the second-ranked candidate can be chosen to be the voter's peak in the societal axis. We assume that the remaining positions in each vote—irrelevant from the point of view of our proof—are filled in in an easily computable way consistent with the societal axis L . For example, each voter we describe below could rank candidates as follows: (a) in the first up to two positions of the vote he or she would rank the candidates as described below (appropriately choosing the candidate to which he or she swoons, and the candidate who takes the role of the voter's peak on the societal axis), (b) in the remaining positions the voter would first rank the remaining candidates “to the left” of the peak and then those “to the right” of the peak.)

1. For each set $S_j \in \mathcal{S}$, for each $b_i \in S_j$, we have $2k$ votes $a_j > b_i > \cdots$.
2. For each set $S_j \in \mathcal{S}$ we have a single vote $a_j > p > \cdots$.
3. We have $2nk + k - n$ voters that rank p first.
4. We have $2nk$ voters that rank d first.
5. For each $b_i \in B$, we have $2nk + 2k - 2k\ell_i$ voters that rank b_i first.

We note that in election (C, V) the scores of candidates are as follows:

1. p has $2nk + k$ points,
2. d has $2nk$ points, and
3. each candidate $b_i \in B$ has $2nk + 2k$ points.

That is, the winners of plurality election (C, V) are exactly the candidates in B . We claim that there is a set A' , $A' \subseteq A$, such that $\|A'\| \leq k$ and p is a winner of plurality election $(C \cup A', V)$ if and only if I is a yes instance of X3C (that is, if there exists a collection of exactly k sets from \mathcal{S} that union to B ; such a collection of sets is called an exact set cover of B).

Let A'' be some subset of A . It is easy to see that in election $(C \cup A'', V)$, plurality scores of candidates are as follows: p has score $2nk + k - \|A''\|$, d has score $2nk$, each candidate $b_i \in B$ has score $2nk + 2k - 2k\|\{a_j \in A'' \mid b_i \in S_j\}\|$, and each candidate $a_i \in A''$ has score $6k + 1$.

Assume that p is a winner of election $(C \cup A'', V)$. Since d 's score is $2nk$ and p 's score is $2nk + k - \|A''\|$, it holds that $\|A''\| \leq k$. Further, for each $b_i \in B$ it holds that b_i 's score is no larger than that of p . It is easy to verify that this is possible only if A'' corresponds to an exact set cover of B (the score of each of $3k$ candidates in B has to be decreased and each $a_j \in A''$ corresponds to decreasing the score of exactly three candidates in B).

On the other hand, if A'' corresponds to an exact cover of B , then p is a winner of election $(C \cup A'', V)$. In such a case $\|A''\| = k$ and so the score of p is $2nk$. Since each $a_j \in A''$ corresponds to a set $S_j \in \mathcal{S}$ that

contains three unique members of B , the score of each $b_i \in B$ is $2nk$. The score of d is $2nk$ as well. Each $a_j \in A''$ has score $6k + 1 < 2nk$ (this is so because $n \geq 4$ and $k \geq 2$). The proof is complete.

We now move on to the case of CCDC. CCDC for plurality over swoon-SP societies is clearly in NP and we focus on proving NP-hardness. We do so by giving a reduction from X3C. Let $I = (B, \mathcal{S})$ be an X3C instance, where $B = \{b_1, \dots, b_{3k}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$. Without loss of generality we assume that $k > 5$. We use societal axis $pLdLb_1L \dots Lb_{3k}La_1L \dots La_n$.

We construct an instance of CCDC for plurality as follows. Set $A = \{a_1, \dots, a_n\}$ and let $E = (C, V)$ be an election, where $C = B \cup A \cup \{p\}$ and V contains the following groups of votes (for each vote we only specify up to two top candidates and up to one ranked-lowest candidate; the reader can verify that using societal axis L it is possible to create swoon-SP votes of the form we require).

1. For each $S_j \in \mathcal{S}$ and for each $b_i \in S_j$, we have one vote $a_j > b_i > \dots > p$.
2. For each $S_j \in \mathcal{S}$, we have one vote $a_j > p > \dots$.
3. For each $b_i \in B$, we have $k - 1$ votes $b_i > \dots > p$.

In this election the candidates have the following scores:

1. p has 0 points,
2. each $b_i \in B$ has $k - 1$ points, and
3. each a_j , $1 \leq j \leq n$, has 4 points (note that $4 < k - 1$).

We claim that it is possible to ensure that p is a winner of this election by deleting at most k candidates if and only if I is a yes instance of X3C.

First, assume that I is a yes instance of X3C and let A' be a subset of A such that $\{S_i \mid a_i \in A'\}$ is an exact cover of B . It is easy to see that p is a plurality winner of election $E' = (C - A', V)$: Compared to E , in E' the score of p increases by k , the score of each $b_i \in B$ increases by 1, and the scores of remaining members of A do not change. Thus, p and all members of B tie for victory.

On the other hand, assume that there exists a set $A'' \subseteq B \cup A$ of candidates, $\|A''\| \leq k$, such that p is a winner of election $E'' = (C - A'', V)$. Since $\|A''\| \leq k$, there are at least $2k$ candidates from B in E'' and so the score of p in E'' has to be at least $k - 1$, to tie with these candidates. However, the only way to increase p 's score to $k - 1$ (or higher) by deleting at most k candidates is to delete $k - 1$ (or more) candidates from A . Yet if we delete exactly $k - 1$ candidates from A , then there is some candidate b_i in the election whose score is at least k . Thus, A'' must contain exactly k candidates from A . Deleting these candidates increases p 's score to be k . To ensure that the scores of the candidates in B do not exceed k , we must ensure that A'' corresponds to an exact cover of B by sets from \mathcal{S} . This completes the proof. \square

By a simple extension of the above proof, we can also show that allowing the number of mavericks to be some root of the input size cannot be handled either.

Theorem 4.8. *For each $\varepsilon > 0$, CCAC and CCDC for plurality elections over I^ε -maverick-SP societies are NP-complete, where I denotes the input size.*

Proof. If $\varepsilon \geq 1$ then all voters can be mavericks and the theorem certainly holds (because CCAC and CCDC are NP-complete for plurality with unrestricted votes). Let us consider the case when ε is strictly between 0

and 1. In this case we can adapt the proof of Theorem 4.7 by including an appropriate number of padding voters.

Let us first handle the case of CCAC. Let (B, \mathcal{S}) be an instance of X3C where $B = \{b_1, \dots, b_{3k}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$, and let (C, A, V, p, K) be the instance of CCAC for plurality produced by the reduction in the proof of Theorem 4.7. Let L be the societal axis used in the proof of Theorem 4.7. By definition, (B, \mathcal{S}) is a yes instance of X3C if and only if (C, A, V, p, K) is a yes instance of CCAC for plurality. However, of course, we have no guarantee that V contains at most I^ε mavericks (with respect to L), where I denotes the input size of (C, A, V, p, K) . Yet it is easy to verify that for each positive integer t , (C, A, V, p, K) is a yes instance of CCAC for plurality if and only if $(C, A, V \cup V'_t, p, K)$ is a yes instance of the same problem, where V'_t is a collection of t blocks of votes that each contain the following $3k + 2$ votes:

1. For each i , $1 \leq i \leq 3k$, there is a single vote that is single-peaked with respect to L and ranks b_i first and p last⁴ (note that, by our choice of L in the proof of Theorem 4.7, such a vote exists).
2. There is a single vote that is single-peaked with respect to L and ranks p first.
3. There is a single vote that is single-peaked with respect to L and ranks d first and p last.

It is easy to see that by choosing a large enough value of t (but polynomially bounded in $I^{\frac{1}{\varepsilon}}$), it is possible to form an instance $(C, A, V \cup V'_t, p, K)$, whose encoding size is I' , that is a yes instance of CCAC for plurality if and only if (B, \mathcal{S}) is a yes instance of plurality, and which contains at most I'^ε mavericks with respect to the societal axis L (namely, the voters in V). This proves our theorem for the CCAC case.

Essentially the same proof approach works for the CCDC case. The crucial observation here is that the proof of the CCDC case of Theorem 4.7 ensures that deleting candidates outside of the set A is never a successful strategy. Adding the voters V'_t does not affect this observation because in all votes in V'_t candidate p is either ranked first or ranked last (however, of course, for the case of CCDC each of the t blocks of votes in V'_t contains only $3k + 1$ votes; the vote that ranks d first and p last is not included, and the candidate d does not occur in any of the other votes). \square

Theorem 4.15. *CCAC and CCDC for plurality elections are NP-complete for $\text{Dodgson}_{m-2}\text{-SP}$ societies and for $\text{PerceptionFlip}_{m-2}\text{-SP}$ societies, where m is the total number of candidates involved in the election.*

It is easy to see that Theorem 4.15 is a simple corollary to the proof of Theorem 4.7. If m is the total number of candidates involved in the election then both in $\text{Dodgson}_{m-2}\text{-SP}$ societies and in $\text{PerceptionFlip}_{m-2}\text{-SP}$ societies the voters can legally rank any two candidates on top of their votes (see lemma below). Further, the societal axis in the CCDC part of the proof of Theorem 4.7 is such that the voters can easily rank p last if need be. This is all that we need for the proof of Theorem 4.7 to work for the case of Theorem 4.15.

Lemma C.3. *Let $C = \{c_1, \dots, c_m\}$ be a set of candidates, $m \geq 2$, and let L be a societal axis over C such that $c_1 L c_2 L \dots L c_m$. For each two candidates $c_i, c_j \in C$, there exist two preference orders of the form $c_i > c_j > \dots$ such that the first is nearly single-peaked in the sense of $\text{Dodgson}_{m-2}\text{-SP}$ societies and the second one is nearly single-peaked in the sense of $\text{PerceptionFlip}_{m-2}\text{-SP}$ societies. Further, if $i \neq 1$ and $j \neq 1$, it is possible to ensure that these preference orders rank c_1 last.*

Proof. Let c_i and c_j be two arbitrary, distinct candidates in C . We first consider the case of $\text{Dodgson}_{m-2}\text{-SP}$ societies. Let $>'$ be an arbitrary preference order that is single-peaked with respect to L and that ranks c_i

⁴For CCAC it is not even necessary to require that p is ranked last. However, we will also use V'_t for the CCDC case, where it is necessary.

first (and c_1 last, if $i \neq 1$ and $j \neq 1$). We obtain $>$ from $>'$ by shifting c_j forward in $>'$ to the second position (that is, just below c_i). It is easy to see that this requires at most $m - 2$ swaps.

For the case of $\text{PerceptionFlip}_{m-2}$ -SP societies, note that by using at most $m - 2$ swaps of adjacent candidates, it is possible to obtain a societal axis L' from L where candidates c_i and c_j are adjacent (and where, if $i \neq 1$ and $j \neq 1$, it still holds that $c_1 L c_k$ for each k , $2 \leq k \leq m$). Clearly, there is a preference order $>$ that is single-peaked with respect to L' and that ranks c_i first and c_j second (and c_1 last, if $i \neq 1$ and $j \neq 1$). \square

C.3 Proof of Theorem 4.14

Theorem 4.14. *For each constant k , CCAC and CCDC for plurality elections are in P for k -local elections.*

We first give a polynomial-time algorithm for CCAC for plurality k -local elections. The main idea of our algorithm is the following. Let p be the candidate whose victory we want to ensure in our input k -local instance of plurality CCAC. We first add up to $2k$ candidates so that the score of p is fixed, and then we run a dynamic programming algorithm that ensures that no candidate has score higher than this fixed score of p . Of course, we do not know which candidates to add in the first part of the algorithm, so we perform an exhaustive search (since k is a constant, it is possible to perform such a search in polynomial time). We will first describe the dynamic programming algorithm in Lemma C.4 and then we will provide the main algorithm. Before we proceed with this plan, we need to provide some additional notation.

Let $E = (C \cup A, V)$ be a k -local plurality election, where we interpret C as the registered candidates and A as the spoiler candidates. Let L be the societal axis for E . We rename the candidates so that $D = C \cup A = \{d_1, \dots, d_m\}$ and $d_1 L d_2 L \dots L d_m$. For each set $B \subseteq D$, we define $\text{lt}(B)$ to be the minimal (leftmost) element of B with respect to L and $\text{rt}(B)$ to be the maximal (rightmost) element of B with respect to L . For each $d_i \in D$ we define $\mathcal{S}(d_i)$ to be the family of sets $\{N(L, C \cup A', d_i, k) \mid A' \subseteq A, d_i \in C \cup A'\}$. The reader can verify that each $\mathcal{S}(d_i)$ contains a number of sets that is at most polynomial in $\|C \cup A\|^k$ and that each $\mathcal{S}(d_i)$ is easily computable (to compute $\mathcal{S}(d_i)$ it suffices to consider sets A' of cardinality at most $2k + 1$).

Lemma C.4. *Let $E = (C \cup A, V)$ be a plurality election, where $C = \{c_1, \dots, c_{m'}\}$ and $A = \{a_1, \dots, a_{m''}\}$, such that E is k -local for some positive integer k . There exists an algorithm that given election E , integer k , societal axis L with respect to which E is k -local, and a nonnegative integer t outputs the cardinality of a smallest (in terms of cardinality) set $A' \subseteq A$ such that the plurality scores of all candidates in election $(C \cup A', V)$ are at most t , or indicates that no such set A' exists. This algorithm runs in time polynomial with respect to $(\|C \cup A\| + \|V\|)^k$.*

Proof. The proof of this lemma is a much extended version of the proof of Lemma 3.7 of [23]. Let the notation be as in the statement of the lemma. We assume that C is nonempty.

We let $D = C \cup A$ and, without loss of generality, we rename the candidates so that $D = \{d_1, \dots, d_m\}$, where $m = m' + m''$, and $d_1 L d_2 L \dots L d_m$. Without loss of generality, we assume that $d_1, d_m \in C$ (if this was not the case, we could extend C to include two additional candidates, ranked last by all voters, without destroying k -locality of the election).

For each $d_i \in D$ and each $D' \in \mathcal{S}(d_i)$ we define $f(d_i, D')$ to be the cardinality of a smallest (with respect to cardinality) set $A' \subseteq A$ such that:

1. For each candidate $d_j \in C \cup A'$ such that $j \leq i$ it holds that $\text{score}_{(C \cup A', V)}(d_j) \leq t$.
2. If $d_{j'} = \text{lt}(D')$ and $d_{j''} = \text{rt}(D')$ then $D' = (C \cup A') \cap \{d_{j'}, d_{j'+1}, \dots, d_{j''}\}$. (Since $d_1, d_m \in C$, this is equivalent to $D' = N(L, C \cup A', d_i, k)$.)

If such an A' does not exist, then we set $f(d_i, D') = \infty$.

Let d_i be a candidate in D and let D' be a member of $\mathcal{S}(d_i)$. Intuitively, D' describes the intended k -radius neighborhood of d_i . Function $f(d_i, D')$ tells us how many candidates from A we need to add to election (C, V) so that in the resulting election the k -radius neighborhood of d_i is exactly D' (which fixes the score of d_i), the score of d_i is at most t , and the scores of candidates preceding d_i (in terms of L) also are at most t .

Since $d_m \in C$, it is easy to verify that our algorithm should output $\min\{f(d_m, D') \mid D' \in \mathcal{S}(d_m)\}$. Thus, in the rest of the proof we describe how to compute f using dynamic programming.

It is easy to see that for each $D' \in \mathcal{S}(d_1)$ it is possible to directly compute the value $f(d_1, D')$. To compute $f(d_i, D')$ for arbitrary $d_i \in D$, $D' \in \mathcal{S}(d_i)$, we use the following, natural to derive, recursive relation. Let us fix some $d_i \in D$, $i > 1$, and $D' \in \mathcal{S}(d_i)$. Let $j = \max\{j' \mid j' < i \text{ and } d_{j'} \in D'\}$. We observe that:

$$f(d_i, D') = \min\{f(d_j, D'') \mid D'' \in \mathcal{S}(d_j) \text{ and } \text{score}_{(D', V)}(d_i) \leq t \text{ and } D' \cup \{\text{lt}(D'')\} = D'' \cup \{\text{rt}(D')\}\} + \chi_A(\text{rt}(D')),$$

where χ_A is the characteristic function of A . (Note that if $\text{rt}(D') \in D''$ for $D'' \in \mathcal{S}(d_j)$, then $\text{rt}(D') = d_m$ and so $\chi_A(\text{rt}(D')) = 0$.) Using standard dynamic programming techniques we can thus compute f in time polynomial in $(\|C \cup A\| + \|V\|)^k$. \square

We now prove that CCAC for plurality k -local elections is in P.

Lemma C.5. *For each fixed k , CCAC for k -local plurality elections, where the societal axis L is given, is in P.*

Proof. Our input instance contains the following elements: (a) an election $E = (C \cup A, V)$, where $C = \{c_1, \dots, c_{m'}\}$ are the registered candidates and $A = \{a_1, \dots, a_{m''}\}$ are the spoiler candidates; (b) a candidate p in C (of course, p is one of the c_i 's but we assign him or her also this special name); (c) a nonnegative integer K ; (d) a societal axis L over $C \cup A$. We ask if there exists a set $A' \subseteq A$ such that:

1. $\|A'\| \leq K$, and
2. p is a winner of plurality election $(C \cup A', V)$.

We rename the candidates so that $D = C \cup A = \{d_1, \dots, d_m\}$, where $m = m' + m''$ and $d_1 L d_2 L \dots L d_m$. Let w be an integer such that $p = d_w$.

The idea of our algorithm is the following: Fix the k -radius neighborhood of p (so that we fix p 's score) and ensure—using the algorithm from Lemma C.4—that the remaining candidates have no more points than p has.

Our algorithm works as follows. For each possible k -radius neighborhood D' of p (i.e., for each $D' \in \mathcal{S}(p)$) we execute the following steps.

1. Set $K_p = \|D' \cap A\|$. (K_p is the number of candidates we need to add to ensure that p has exactly k -radius neighborhood D' .)
2. Set $t = \text{score}_{(D', V)}(p)$ (by E 's k -locality, t is the score of p in any election where the k -radius neighborhood of p is D').
3. Check how many candidates are needed to ensure that candidates “to the left” of p do not beat p):
 - (a) Set j' to be such that $d_{j'} = \text{lt}(D')$.

- (b) Set $C_{left} = (\{d_1, \dots, d_{j'-1}\} \cap C) \cup D'$ and set $A_{left} = \{d_1, \dots, d_{j'-1}\} \cap A$. (C_{left} is the set of all registered candidates “to the left” of D' , union D' (we treat D' as already fixed); A_{left} is the set of spoiler candidates to the left of D' .)
 - (c) Using Lemma C.4 compute the minimal number of candidates from A_{left} that need to be added to election (C_{left}, V) so that each candidate in the resulting election has score at most t . Call this number K_{left} . If it is impossible to achieve the desired effect, drop this D' .
4. Check how many candidates are needed to ensure that candidates “to the right” of p do not beat p):
- (a) Set j'' to be such that $d_{j''} = \text{rt}(D')$.
 - (b) Set $C_{right} = (\{d_{j''+1}, \dots, d_m\} \cap C) \cup D'$ and set $A_{right} = \{d_{j''+1}, \dots, d_m\} \cap A$. (C_{right} is the set of all registered candidates “to the right” of D' , union D' (we treat D' as already fixed); A_{right} is the set of spoiler candidates “to the right” of D' .)
 - (c) Using Lemma C.4 compute the minimal number of candidates from A_{right} that need to be added to election (C_{right}, V) so that each candidate in the resulting election has score at most t . Call this number K_{right} . If it is impossible to achieve the desired effect, drop this D' .
5. If $K_p + K_{left} + K_{right} \leq K$ then accept.

If the above procedure does not accept for any D' then reject.

By Lemma C.4 and the fact that k is a fixed constant, it is easy to see that this algorithm works in polynomial time. The correctness is easy to observe as well. \square

We now move on to the case of CCDC for plurality k -local elections.

Lemma C.6. *For each fixed k , CCDC for k -local plurality elections, where the societal axis L is given, is in P.*

Proof. Let $E = (C, V)$ be our input election, p be the preferred candidate, and K be a nonnegative integer. Our goal is to determine if it is possible to ensure that p is a winner by deleting at most K candidates. Let L be the input societal axis with respect to which E is k -local.

We rename the candidates in C so that $C = \{\ell_{m'}, \dots, \ell_1, p, r_1, \dots, r_{m''}\}$ and $\ell_{m'} L \cdots L \ell_1 L p L r_1 L \cdots L r_{m''}$. Recall that by definition of k -local plurality elections, the score of p depends only on the presence of k candidates “to the left of p ” and k candidates “to the right of p ” (with respect to L). Our algorithm works as follows:

1. For each size- $\min(k, m')$ subset L of $\{\ell_1, \dots, \ell_{m'}\}$ and each size- $\min(k, m'')$ subset R of $\{r_1, \dots, r_{m''}\}$ execute the following steps:
 - (a) Let i be the largest integer such that $\ell_i \in L$ and let j be the largest integer such that $r_j \in R$.
 - (b) Let $D = \{\ell_i, \dots, \ell_1, r_1, \dots, r_j\} - (L \cup R)$ (at this point, intuitively, D is the unique smallest set of candidates that one has to delete from C to ensure that the k -radius neighborhood of p is exactly $L \cup R$).
 - (c) Execute the following loop: If there is a candidate $c \in C - D$, $c \neq p$, such that the score of c in $(C - D, V)$ is higher than that of p , then add c to D .
 - (d) If $\|D\| \leq K$ then accept.
2. Reject.

Since k is a constant, there are only polynomially many pairs of sets L and R to try. Thus, it is easy to see that the algorithm runs in polynomial time. To see the correctness, it suffices to note the following two facts. First, the score of p depends only on the k -radius neighborhood of p . Second, it is impossible to decrease a score of a candidate by deleting (other) candidates, so if for a given k -radius neighborhood of p some candidates still have score higher than p , the only way to ensure that they do not preclude p from winning is by deleting them.⁵ \square

C.4 Proof of Theorem 4.16

The following lemma will be very useful in proving Theorem 4.16.

Lemma C.7. *Let $E = (C, V)$ be an election where $C = \{c_1, \dots, c_m\}$ and where voters in V are single-caved with respect to societal axis $c_1 L c_2 L \dots L c_m$. Then each voter in V ranks first either c_1 or c_m .*

Proof. Assume for the sake of contradiction that there is $c_i \in C$, $i \notin \{1, m\}$, such that some voter v in V ranks c_i first. However, it holds that $c_1 L c_i L c_m$. Since, by assumption, v prefers c_i to c_1 , by definition of single-cavedness it holds that v prefers c_m to c_i . This is a contradiction. \square

With Lemma C.7 in hand, we can now prove Theorem 4.16.

Theorem 4.16. *CCAC and CCDC for plurality elections are in P for single-caved societies.*

Proof. Let us consider the CCAC case first. Let (C, A, V, p, K) be our input instance of CCAC for plurality, where votes in V are single-caved with respect to a given societal axis L . We assume that $\|C\| \geq 2$ (otherwise p , the only candidate, is already a winner).

Let a be some candidate in A . We claim that if p is not a winner of election $E = (C, V)$ then p is not a winner of election $E' = (C \cup \{a\}, V)$. First, adding a cannot increase p 's plurality score. Thus, if p 's plurality score in E is 0 then it is 0 in E' as well and p is not a winner in either of them. Thus, let us assume that there is at least one voter that prefers p to all other candidates in C . This means that we can assume, without loss of generality, that there is a candidate $d \in C$ such that for each candidate $c \in C - \{p, d\}$ it holds that $p L c L d$. By Lemma C.7, p and d are the only candidates in election E whose plurality score is nonzero. We assume that p does not win in E , so the score of p is smaller than the score of d . We now consider three possible cases, depending on a 's position on the societal axis.

1. If $a L p$ then it is easy to note that in every vote in which p was ranked first prior to adding a , now a is ranked first, and so p is not a winner of the election.
2. If $d L a$ then a is ranked first in each vote in which d was ranked first prior to adding a , and so now p loses to a .
3. If $p L a L d$ then adding a to the election does not change plurality scores of p and d and thus p still loses to d .

Thus, by induction on the number of added candidates, it is impossible to move p from losing an election to winning it by adding candidates. Our CCAC algorithm simply checks if p is a winner already, accepts if so and rejects otherwise.

⁵Note that in the process of doing so we might change the k -radius neighborhood of p , but that does not affect the correctness of the algorithm.

Let us now consider the case of CCDC for plurality and single-caved societies. Let (C, V, p, K) be our input instance where votes in V are single-caved with respect to given societal axis L . Let us rename the candidates so that $C = \{c_1, \dots, c_m\}$, $c_1 L \dots L c_m$, and let us fix i such that $p = c_i$. Let $\mathcal{D} = \{\{c_1, c_2, \dots, c_{i-1}, c_j, \dots, c_m\} \mid j > i\} \cup \{\{c_1, \dots, c_k, c_{i+1}, c_{i+2}, \dots, c_m\} \mid k < i\}$. By Lemma C.7 and the definition of single-cavedness, it is easy to see that p can become a winner of election (C, V) by deleting at most K candidates if and only if $V = \emptyset$ or there is a set D in \mathcal{D} such that $\|D\| \leq K$ and p is a winner of election $(C - D, V)$. \square

D Proof from Section 5

We provide the proof of Section 5’s theorem.

Theorem 5.1. *Bribery, negative-bribery, and strongnegative-bribery for approval elections over log-maverick-SP societies are each in P, in both the standard and the marked model.*

Proof. We first prove in detail the “bribery” case (the first of the three types of bribery that the theorem covers), in both its marked-model and standard-model cases.

Let us look first at the marked model. In this case, much as in the proof of Theorem 4.1, we will note that an earlier paper is implicitly obtaining a stronger result than what its theorem states, and then we will use that observation to build a disjunctive truth-table reduction from our problem to that problem. In this case, the earlier paper is not Faliszewski et al. [23] as it was in Theorem 4.1, but rather is Brandt et al. [4]. The result of theirs that we focus on is their theorem stating that approval bribery is in polynomial time for single-peaked societies. This is “Theorem 4” of Brandt et al. [4], but for its proof/algorithm, one needs to refer to Appendix A.2 of that paper’s technical report version [5]. Now, by inspection of that proof, one can see that the algorithm given there does not need all the voters it operates on to respect the societal ordering. Rather, it can handle perfectly well the case where each voter has an “open to bribes” flag, and every voter whose open to bribes flag is set respects the single-peaked ordering. (In that algorithm, as modified to handle this, the surpluses are computed with respect to all the voters—both those with the flag set and those with the flag unset. But then the pool of voters that the algorithm looks at to try to find a good bribe is limited to just those with the open to bribes flag set, although the surplus recomputations throughout the algorithm are always with respect to the entire set of voters. This is a slight extension of the algorithm, but is clearly correct, for the same reasons the original algorithm is.) Note that the algorithm can bribe voters whose open to bribes flag is set, but (by the nature of the algorithm) will only bribe them to values consistent with the societal order. Call the language problem defined by this FlagBribe.

Having made the previous paragraph’s observation, we can now disjunctive truth-table reduce to FlagBribe—which is put into polynomial time by the above algorithm (due to Brandt et al. [4], except slightly adapted as just mentioned). We do so as follows. Suppose our logarithmic bound is given. Given an input to our problem, we check that the number of voters with the maverick-enabled flag (not to be confused with the open to bribes flag mentioned above) set does not exceed the logarithmic bound; if it does, reject immediately. Otherwise, for each of member A of the powerset of the set of maverick-enabled voters (i.e., for each choice of which of the maverick-enabled voters we will bribe), we will generate at most one instance of FlagBribe as follows. If $\|A\| > K$, generate no instance. (The number of voters being bribed would exceed the problem’s bound.) Otherwise, generate an instance of FlagBribe that is the same set of voters as our instance, except with the members of A modified to each approve of p and only p . The voters who in our original problem were not maverick-enabled will all have their open to bribes flag set. All others will have their open to bribes flag unset. Set K to now be $K - \|A\|$.

So, since there are a logarithmic number of maverick-enabled voters, the powerset above is polynomial in size, and we generate a polynomial number of (polynomial-sized) instances of FlagBribe. It is clear that our original problem has a successful bribe exactly if at least one of those instances has a successful bribe (i.e., belongs to FlagBribe). This is so, due to the properties of the algorithm underlying FlagBribe, and the fact that if there is a bribe of K voters that makes p a winner, then the same bribe action except with any subset of them instead bribed to approve only of p will also make p a winner. Changing a voter to approve of p and only p is a best possible bribe of that voter, if the voter will be bribed at all.

That concludes the marked model case for bribery. We turn now to the standard model case. In this case, each voter can potentially turn into a maverick. And so with an $\mathcal{O}(\log(\text{ProblemInputSize}))$ bound on the number of mavericks, as long as the bribery problem itself has a generously large K , to even decide which voters to make into mavericks would seem to involve $\binom{\|C\|}{\mathcal{O}(\log(\text{ProblemInputSize}))}$ options—superpolynomially many, which potentially is a worry if they can be turned into complex mavericks. But we are again saved here by the fact that any good bribe of a voter is at least as good if one just bribes that voter to approve only p (and clearly that vote is also inherently consistent with the societal ordering). That means that if there is a good set of bribes, then there is a good set of bribes that never bribes people to vote in ways that are inconsistent with the societal order. But given that, we can turn this case into our marked-model case. (The calls to FlagBribe that will underlie the handling of that case indeed also may present problems with superpolynomial numbers of options as to which voters to bribe. But due to the single-peakedness-respecting nature of all voters who are open to bribes, that can be handled easily—that is the real power of FlagBribe’s underlying algorithm: it uses single-peakedness to tame combinatorial explosion.)

In particular, we can proceed here as follows. Take our input. Reject if the number of voters who are inconsistent with the societal ordering conflicts with our logarithmic bound. Otherwise, have the maverick-enabled flag be set for each voter who violates the societal ordering and have the maverick-enabled flag be unset for all other voters. Keep the K parameter the same as it originally was. And then solve this marked-model case as described above. This works, due to the comments of the previous paragraph.

That covers in detail the case of bribery. The remaining two cases, negative-bribery and strongnegative-bribery, are similarly proven by noting that one can alter the algorithms from Brandt et al. [4] for those two cases, and by noting that if there is a good bribe in these models, then there is a good bribe where no bribed voter will have any approval set other than either “just p ” or the empty set. \square